Exam MSBA STA380

Rushiil Deshmukh

08/01/2021

Chapter 2 Problem 10

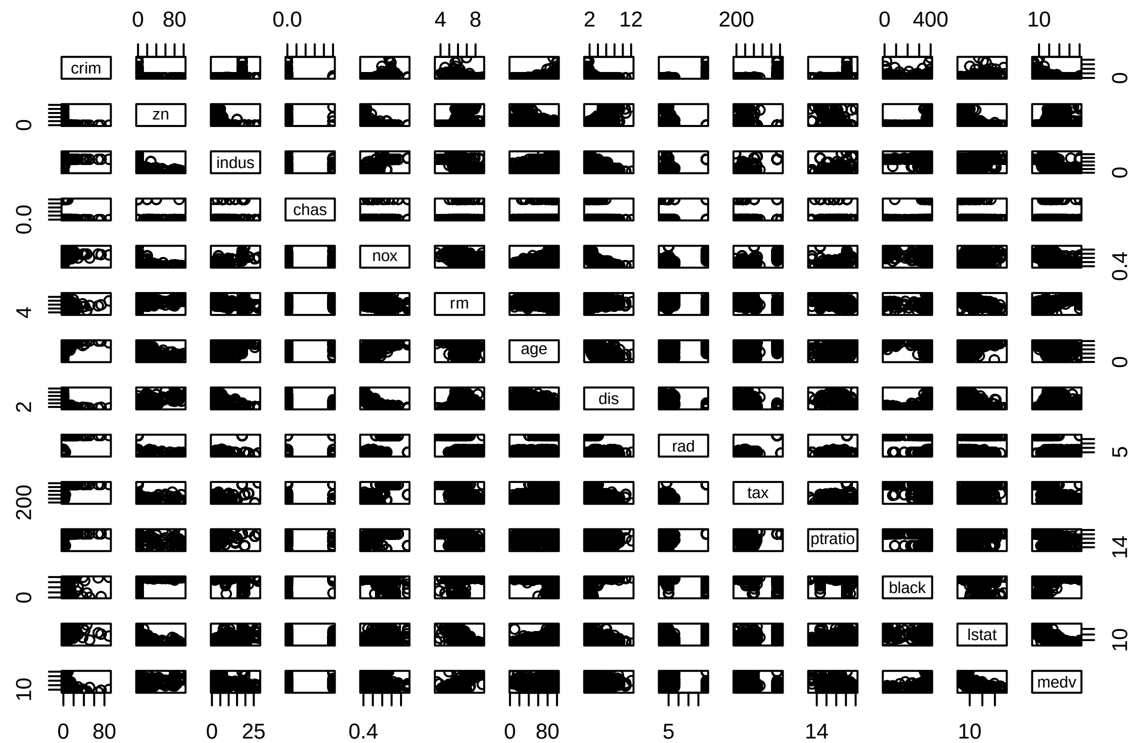
Part A

```
## [1] 506  14
```

The Dataset has 506 rows and 14 columns

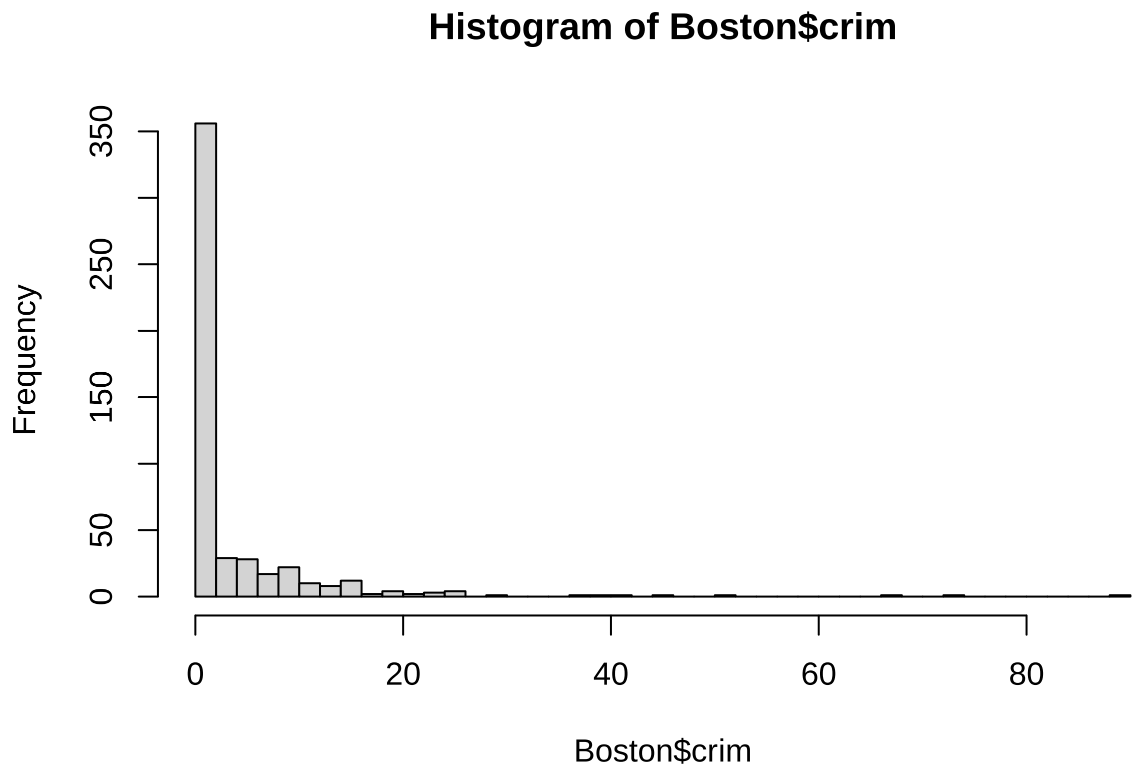
The Variables are:
1. crim - per capita crime rate by town.
2. zn - Proportion of residential land zoned for lots over 25,000 sq.ft.
3. indus - Proportion of non-retail business acres per town.
4. chas - Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
5. nox - Nitrogen oxides concentration (parts per 10 million).
6. rm - Average number of rooms per dwelling.
7. age - Proportion of owner-occupied units built prior to 1940.
8. dis - Weighted mean of distances to five Boston employment centres.
9. rad - Index of accessibility to radial highways. 10. tax - Full-value property-tax rate per $10,000.
11. ptratio - Pupil-teacher ratio by town.
12. black - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town.
13. lstat - Lower status of the population (percent).
14. medv - Median value of owner-occupied homes in $1000s.

Part B



1. Higher tax for homes closer to radial highways (better connectivity)
2. Negative correlation between dis and nox i.e lower NO2 air near city centres
3. Higher ptratio indicates higher medv i.e Better neighborhoods have focused education
4. High crime in areas with high lstat i.e poorer neighborhoods
5. High rm indicates a higher medv due to bigger houses
6. Negative correlation between black and dis

Part C

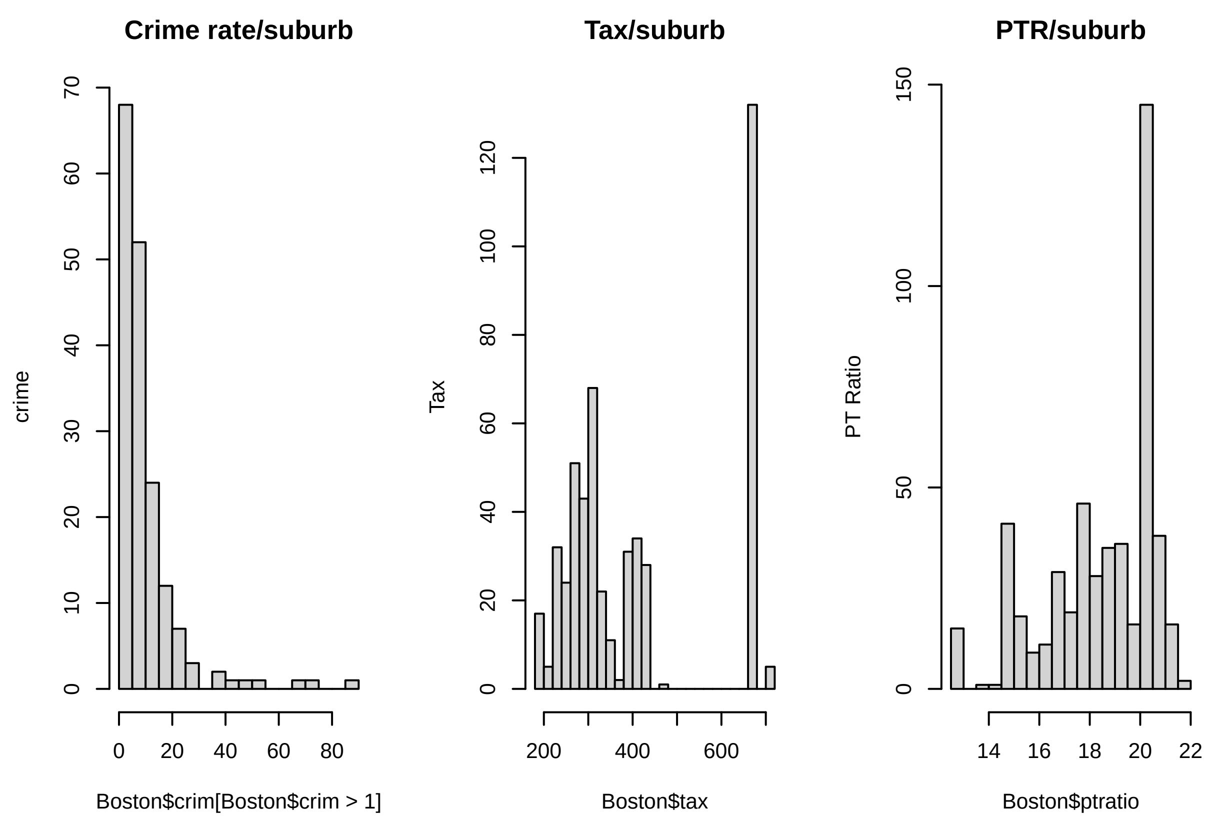



```
##                crim          zn       indus         chas         nox
## crim     1.00000000 -0.20046922  0.40658341 -0.055891582  0.42097171
## zn      -0.20046922  1.00000000 -0.53382819 -0.042696719 -0.51660371
## indus    0.40658341 -0.53382819  1.00000000  0.062938027  0.76365145
## chas    -0.05589158 -0.04269672  0.06293803  1.000000000  0.09120281
## nox      0.42097171 -0.51660371  0.76365145  0.091202807  1.00000000
## rm      -0.21924670  0.31199059 -0.39167585  0.091251225 -0.30218819
## age      0.35273425 -0.56953734  0.64477851  0.086517774  0.73147010
## dis     -0.37967009  0.66440822 -0.70802699 -0.099175780 -0.76923011
## rad      0.62550515 -0.31194783  0.59512927 -0.007368241  0.61144056
## tax      0.58276431 -0.31456332  0.72076018 -0.035586518  0.66802320
## ptratio  0.28994558 -0.39167855  0.38324756 -0.121515174  0.18893268
## black   -0.38506394  0.17552032 -0.35697654  0.048788485 -0.38005064
## lstat    0.45562148 -0.41299457  0.60379972 -0.053929298  0.59087892
## medv    -0.38830461  0.36044534 -0.48372516  0.175260177 -0.42732077
##                  rm         age         dis          rad         tax    ptratio
## crim    -0.21924670  0.35273425 -0.37967009  0.625505145  0.58276431  0.2899456
## zn       0.31199059 -0.56953734  0.66440822 -0.311947826 -0.31456332 -0.3916785
## indus   -0.39167585  0.64477851 -0.70802699  0.595129275  0.72076018  0.3832476
## chas     0.09125123  0.08651777 -0.09917578 -0.007368241 -0.03558652 -0.1215152
## nox     -0.30218819  0.73147010 -0.76923011  0.611440563  0.66802320  0.1889327
## rm       1.00000000 -0.24026493  0.20524621 -0.209846668 -0.29204783 -0.3555015
## age     -0.24026493  1.00000000 -0.74788054  0.456022452  0.50645559  0.2615150
## dis      0.20524621 -0.74788054  1.00000000 -0.494587930 -0.53443158 -0.2324705
```

```
## rad     -0.20984667  0.45602245 -0.49458793  1.000000000  0.91022819  0.4647412
## tax     -0.29204783  0.50645559 -0.53443158  0.910228189  1.00000000  0.4608530
## ptratio -0.35550149  0.26151501 -0.23247054  0.464741179  0.46085304  1.0000000
## black    0.12806864 -0.27353398  0.29151167 -0.444412816 -0.44180801 -0.1773833
## lstat   -0.61380827  0.60233853 -0.49699583  0.488676335  0.54399341  0.3740443
## medv     0.69535995 -0.37695457  0.24992873 -0.381626231 -0.46853593 -0.5077867
##               black      lstat       medv
## crim    -0.38506394  0.4556215 -0.3883046
## zn       0.17552032 -0.4129946  0.3604453
## indus   -0.35697654  0.6037997 -0.4837252
## chas     0.04878848 -0.0539293  0.1752602
## nox     -0.38005064  0.5908789 -0.4273208
## rm       0.12806864 -0.6138083  0.6953599
## age     -0.27353398  0.6023385 -0.3769546
## dis      0.29151167 -0.4969958  0.2499287
## rad     -0.44441282  0.4886763 -0.3816262
## tax     -0.44180801  0.5439934 -0.4685359
## ptratio -0.17738330  0.3740443 -0.5077867
## black    1.00000000 -0.3660869  0.3334608
## lstat   -0.36608690  1.0000000 -0.7376627
## medv     0.33346082 -0.7376627  1.0000000
```

Most neighborhoods are crime free

Part D



Part E

```
## [1] 35
```

35 Suburbs are bound to the Charles River

Part F

```
## [1] 19.05
```

The Median Pupil Teacher Ratio is 19.05

Part G

```
##               399      406
## crim      38.3518  67.9208
## zn         0.0000   0.0000
## indus     18.1000  18.1000
## chas       0.0000   0.0000
## nox        0.6930   0.6930
## rm         5.4530   5.6830
## age      100.0000 100.0000
## dis        1.4896   1.4254
## rad       24.0000  24.0000
## tax      666.0000 666.0000
## ptratio   20.2000  20.2000
## black    396.9000 384.9700
## lstat     30.5900  22.9800
## medv       5.0000   5.0000
```

Suburb number 5 has the lowest median housing value

Part H

Greater than 7 rooms

```
## [1] 64
```

Greater than 8 rooms

```
## [1] 13
```

Greater than 8 rooms - Summary

```
##       crim               zn            indus             chas       
##  Min.   :0.02009   Min.   : 0.00   Min.   : 2.680   Min.   :0.0000  
##  1st Qu.:0.33147   1st Qu.: 0.00   1st Qu.: 3.970   1st Qu.:0.0000  
##  Median :0.52014   Median : 0.00   Median : 6.200   Median :0.0000  
##  Mean   :0.71879   Mean   :13.62   Mean   : 7.078   Mean   :0.1538  
```

```
##  3rd Qu.:0.57834   3rd Qu.:20.00   3rd Qu.: 6.200   3rd Qu.:0.0000
##  Max.   :3.47428   Max.   :95.00   Max.   :19.580   Max.   :1.0000
##       nox               rm             age              dis
##  Min.   :0.4161   Min.   :8.034   Min.   : 8.40   Min.   :1.801
##  1st Qu.:0.5040   1st Qu.:8.247   1st Qu.:70.40   1st Qu.:2.288
##  Median :0.5070   Median :8.297   Median :78.30   Median :2.894
##  Mean   :0.5392   Mean   :8.349   Mean   :71.54   Mean   :3.430
##  3rd Qu.:0.6050   3rd Qu.:8.398   3rd Qu.:86.50   3rd Qu.:3.652
##  Max.   :0.7180   Max.   :8.780   Max.   :93.90   Max.   :8.907
##       rad              tax           ptratio          black
##  Min.   : 2.000   Min.   :224.0   Min.   :13.00   Min.   :354.6
##  1st Qu.: 5.000   1st Qu.:264.0   1st Qu.:14.70   1st Qu.:384.5
##  Median : 7.000   Median :307.0   Median :17.40   Median :386.9
##  Mean   : 7.462   Mean   :325.1   Mean   :16.36   Mean   :385.2
##  3rd Qu.: 8.000   3rd Qu.:307.0   3rd Qu.:17.40   3rd Qu.:389.7
##  Max.   :24.000   Max.   :666.0   Max.   :20.20   Max.   :396.9
##      lstat           medv
##  Min.   :2.47   Min.   :21.9
##  1st Qu.:3.32   1st Qu.:41.7
##  Median :4.14   Median :48.3
##  Mean   :4.31   Mean   :44.2
##  3rd Qu.:5.12   3rd Qu.:50.0
##  Max.   :7.44   Max.   :50.0
```

Crime rates in these suburbs are low, Median housing value is very high and tax is higher too

Chapter 8 Problem 8

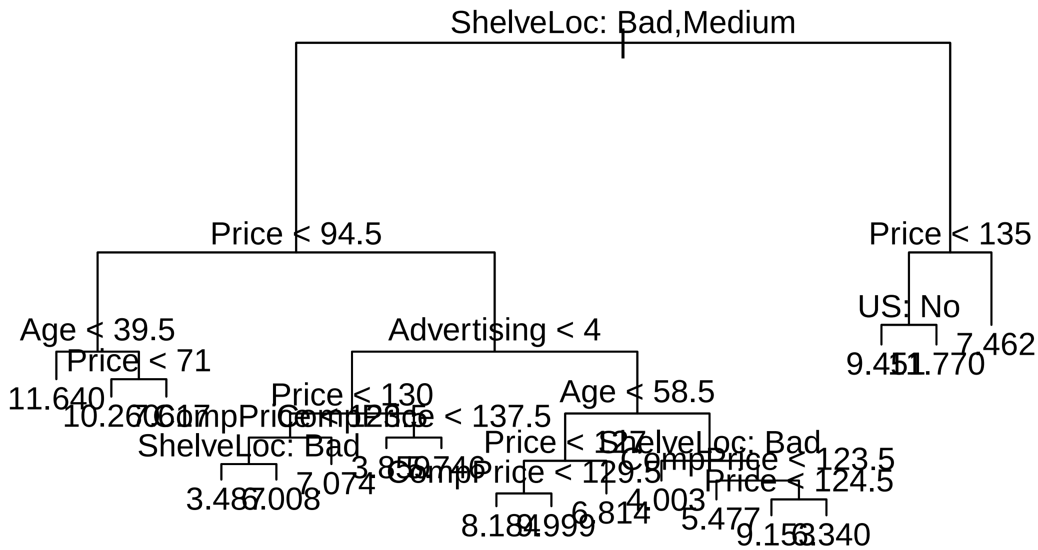
Part A

```
## [1] 200  11
```

Dimensions of the training set are: 200 rows, 11 columns

Part B

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = c_train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Advertising" "CompPrice"
## [6] "US"
## Number of terminal nodes:  18
## Residual mean deviance:  2.167 = 394.3 / 182
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.88200 -0.88200 -0.08712  0.00000  0.89590  4.09900
```




```
## [1] 4.922039
```

ShelveLoc is the primary division variable, therefore the most important - variable. Followed by price. Therefore, if the Shelf location is bad, the sales decrease because the part on the shelf may be deep in a warehouse

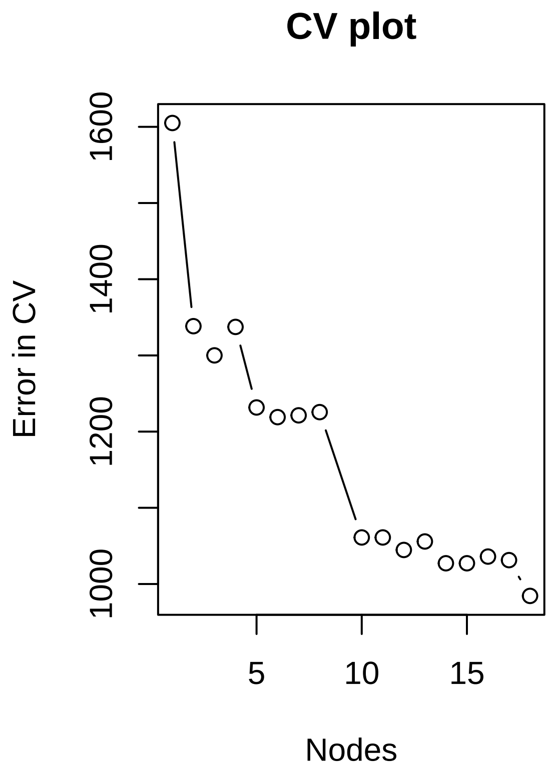
The test MSE is 4.922

Part C

Cross Validation

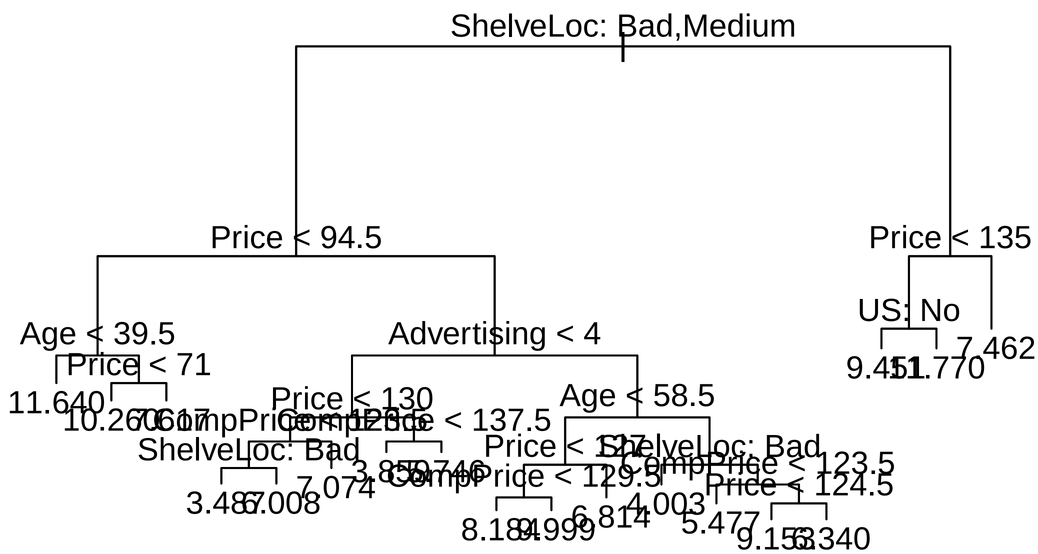
```
## $size
##  [1] 18 17 16 15 14 13 12 11 10  8  7  6  5  4  3  2  1
##
## $dev
##  [1]  984.3936 1031.3372 1036.0021 1027.2166 1027.2166 1055.8168 1044.6955
##  [8] 1061.0899 1061.0899 1225.5973 1221.3487 1219.0219 1231.6886 1337.3952
## [15] 1300.0524 1338.3702 1605.0221
##
## $k
##  [1]      -Inf  16.99544  20.56322  25.01730  25.57104  28.01938  30.36962
##  [8]  31.56747  31.80816  40.75445  44.44673  52.57126  76.21881  99.59459
## [15] 116.69889 159.79501 337.60153
##
```

```
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```



Optimal level of tree nodes is 18, lowest $dev = 984, Pruning won't help

Pruning




```
## [1] 4.922039
```

Cross Validation MSE is 4.922 Cross Validation proves that pruning the tree doesn't change the MSE

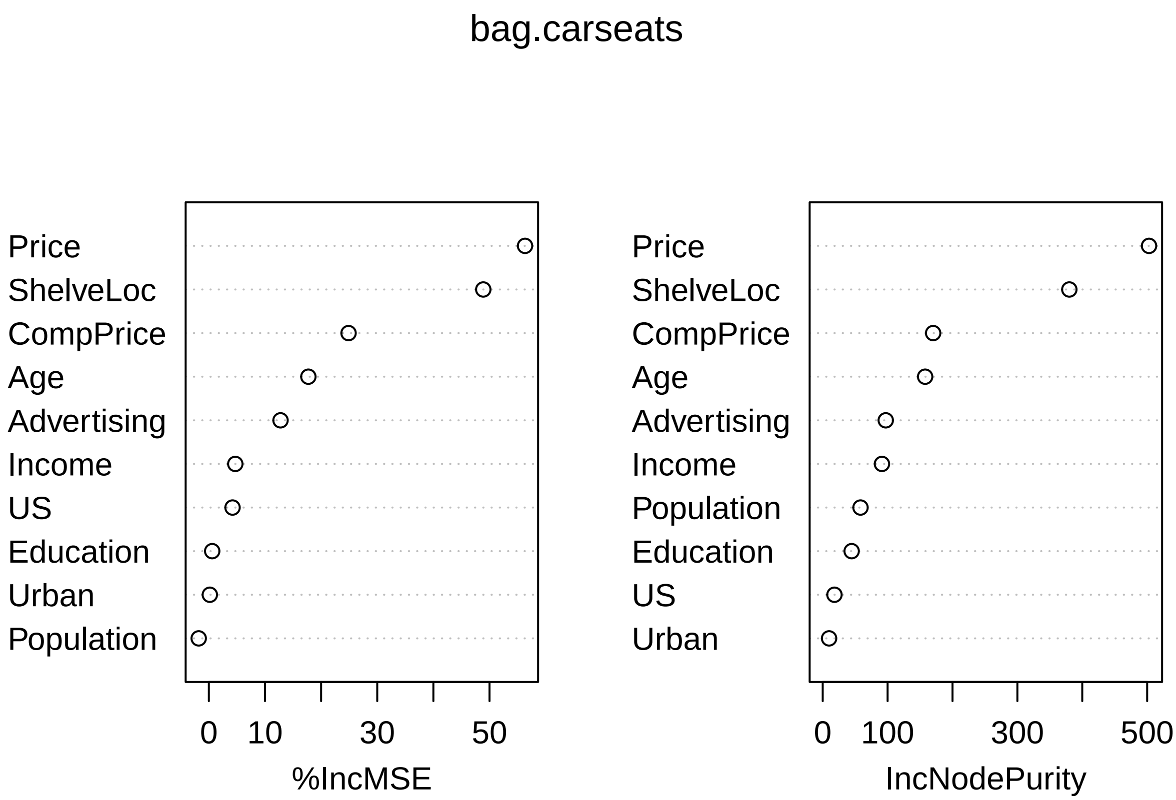
Part D

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
## [1] 2.605253
```

```
##                %IncMSE IncNodePurity
## CompPrice   24.8888481    170.182937
## Income       4.7121131     91.264880
## Advertising 12.7692401     97.164338
## Population  -1.8074075     58.244596
## Price       56.3326252    502.903407
## ShelveLoc   48.8886689    380.032715
## Age         17.7275460    157.846774
## Education    0.5962186     44.598731
## Urban        0.1728373      9.822082
## US           4.2172102     18.073863
```



Price, Shelf Location and CompPrice are the three most important variables Bagging improves the test MSE, bringing it down to 2.6505253

Part E

m = 1

```
## [1] 4.747522
```

Test MSE is 4.7475

m = 2

```
## [1] 3.401523
```

Test MSE is 3.4015

m = sqrt(p)

```
## [1] 2.960559
```

```
##                %IncMSE IncNodePurity
## CompPrice   14.8840765     158.82956
## Income       4.3293950     125.64850
```

```
## Advertising  8.2215192     107.51700
## Population  -0.9488134      97.06024
## Price       34.9793386     385.93142
## ShelveLoc   34.9248499     298.54210
## Age         14.3055912     178.42061
## Education    1.3117842      70.49202
## Urban       -1.2680807      17.39986
## US           6.1139696      33.98963
```

Random Forest increases MSE. Price and Shelf Location are almost equally influential on MSE

Chapter 8 Problem 11

Part A

```
## Loaded gbm 2.1.8
```

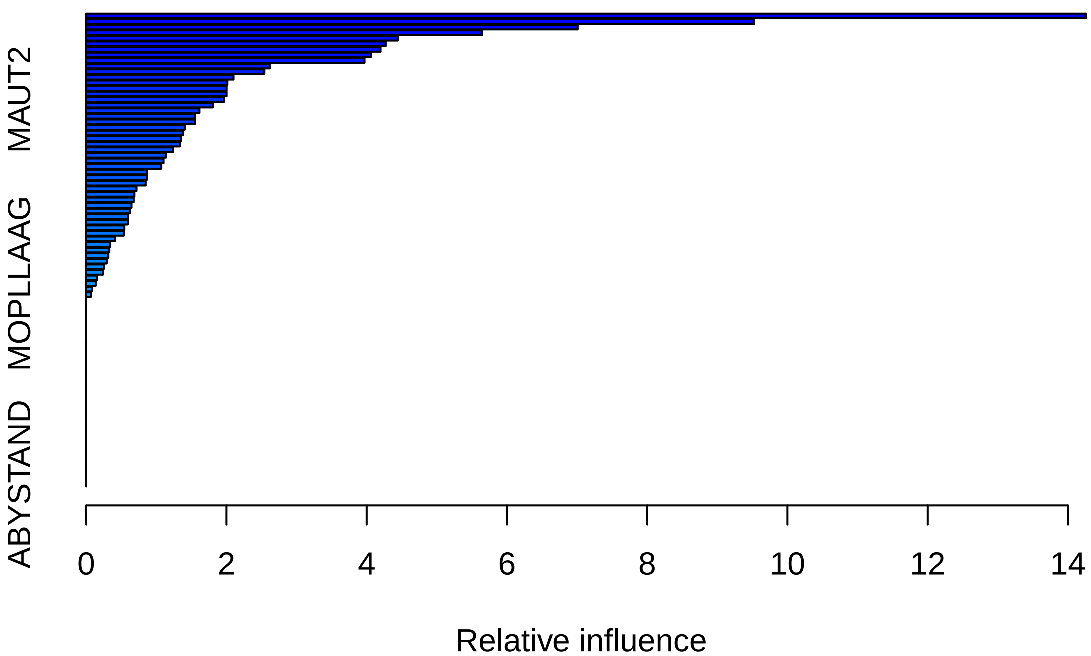
```
## [1] 4822   86
```

Dimensions of the training set are: 4822 rows, 86 columns

Part B

```
## Warning in gbm.fit(x = x, y = y, offset = offset, distribution = distribution, :
## variable 50: PVRAAUT has no variation.
```

```
## Warning in gbm.fit(x = x, y = y, offset = offset, distribution = distribution, :
## variable 71: AVRAAUT has no variation.
```




```
##                 var     rel.inf
## PPERSAUT PPERSAUT 14.25950488
## MKOOPKLA MKOOPKLA  9.52689029
## MOPLHOOG MOPLHOOG  7.00889676
## MBERMIDD MBERMIDD  5.64514340
## ABRAND     ABRAND  4.44285794
## MGODGE     MGODGE  4.27078186
## MOSTYPE   MOSTYPE  4.19761401
## MINK3045 MINK3045  4.05781540
## PBRAND     PBRAND  3.96947092
## PWAPART   PWAPART  2.61977805
## MAUT1       MAUT1  2.54109119
## MGODPR     MGODPR  2.10097241
## MSKA         MSKA  2.01358346
## MSKC         MSKC  2.00255151
## MGODOV     MGODOV  2.00233109
## MAUT2       MAUT2  1.96580896
## MBERARBG MBERARBG  1.80696693
## PBYSTAND PBYSTAND  1.61496285
## MSKB1       MSKB1  1.55227111
## MINKGEM   MINKGEM  1.54983878
## MBERHOOG MBERHOOG  1.40562721
## MFWEKIND MFWEKIND  1.38378851
## MINKM30   MINKM30  1.35369721
## MINK7512 MINK7512  1.33837892
## MOPLMIDD MOPLMIDD  1.23858835
```

```
## MGODRK     MGODRK  1.13897367
## MFGEKIND MFGEKIND  1.10334166
## MRELOV     MRELOV  1.07119101
## MAUT0       MAUT0  0.86937178
## MINK4575 MINK4575  0.86449137
## MRELGE     MRELGE  0.84818329
## MBERBOER MBERBOER  0.71759647
## MOSHOOFD MOSHOOFD  0.68740878
## APERSAUT APERSAUT  0.67746507
## MGEMLEEF MGEMLEEF  0.64654305
## MHKOOP     MHKOOP  0.62155045
## MGEMOMV   MGEMOMV  0.59535646
## MZPART     MZPART  0.59337559
## MZFONDS   MZFONDS  0.54408388
## MHHUUR     MHHUUR  0.53678875
## MBERARBO MBERARBO  0.40844956
## MINK123M MINK123M  0.34332918
## MSKB2       MSKB2  0.32962031
## MSKD         MSKD  0.31548620
## MRELSA     MRELSA  0.29243994
## MFALLEEN MFALLEEN  0.25043557
## PMOTSCO   PMOTSCO  0.23872408
## PLEVEN     PLEVEN  0.15460531
## MOPLLAAG MOPLLAAG  0.13567978
## MBERZELF MBERZELF  0.07978890
## MAANTHUI MAANTHUI  0.06650788
## PWABEDR   PWABEDR  0.00000000
## PWALAND   PWALAND  0.00000000
## PBESAUT   PBESAUT  0.00000000
## PVRAAUT   PVRAAUT  0.00000000
## PAANHANG PAANHANG  0.00000000
## PTRACTOR PTRACTOR  0.00000000
## PWERKT     PWERKT  0.00000000
## PBROM       PBROM  0.00000000
## PPERSONG PPERSONG  0.00000000
## PGEZONG   PGEZONG  0.00000000
## PWAOREG   PWAOREG  0.00000000
## PZEILPL   PZEILPL  0.00000000
## PPLEZIER PPLEZIER  0.00000000
## PFIETS     PFIETS  0.00000000
## PINBOED   PINBOED  0.00000000
## AWAPART   AWAPART  0.00000000
## AWABEDR   AWABEDR  0.00000000
## AWALAND   AWALAND  0.00000000
## ABESAUT   ABESAUT  0.00000000
## AMOTSCO   AMOTSCO  0.00000000
## AVRAAUT   AVRAAUT  0.00000000
## AAANHANG AAANHANG  0.00000000
## ATRACTOR ATRACTOR  0.00000000
## AWERKT     AWERKT  0.00000000
## ABROM       ABROM  0.00000000
## ALEVEN     ALEVEN  0.00000000
## APERSONG APERSONG  0.00000000
## AGEZONG   AGEZONG  0.00000000
```

```
## AWAOREG   AWAOREG  0.00000000
## AZEILPL   AZEILPL  0.00000000
## APLEZIER APLEZIER  0.00000000
## AFIETS     AFIETS  0.00000000
## AINBOED   AINBOED  0.00000000
## ABYSTAND ABYSTAND  0.00000000
```

```
## [1] 0.05419263
```

MSE is 0.0542 PPERSAUT and MKOOPKL are the most impactful variables

Part C

Prediction

```
##    pred.test
##        0    1
##   0 4499   34
##   1  280    9
```

Fraction of people that might make a purchase is 0.2093 or 20.93%

KNN

```
## [1] 0.058
```

MSE is 0.058 It is apparent that the MSE value using knn is higher than that using boosted trees, therefore the boosting model fits the data better than k nearest neighbours.

Chapter 3 Problem 15

Part A

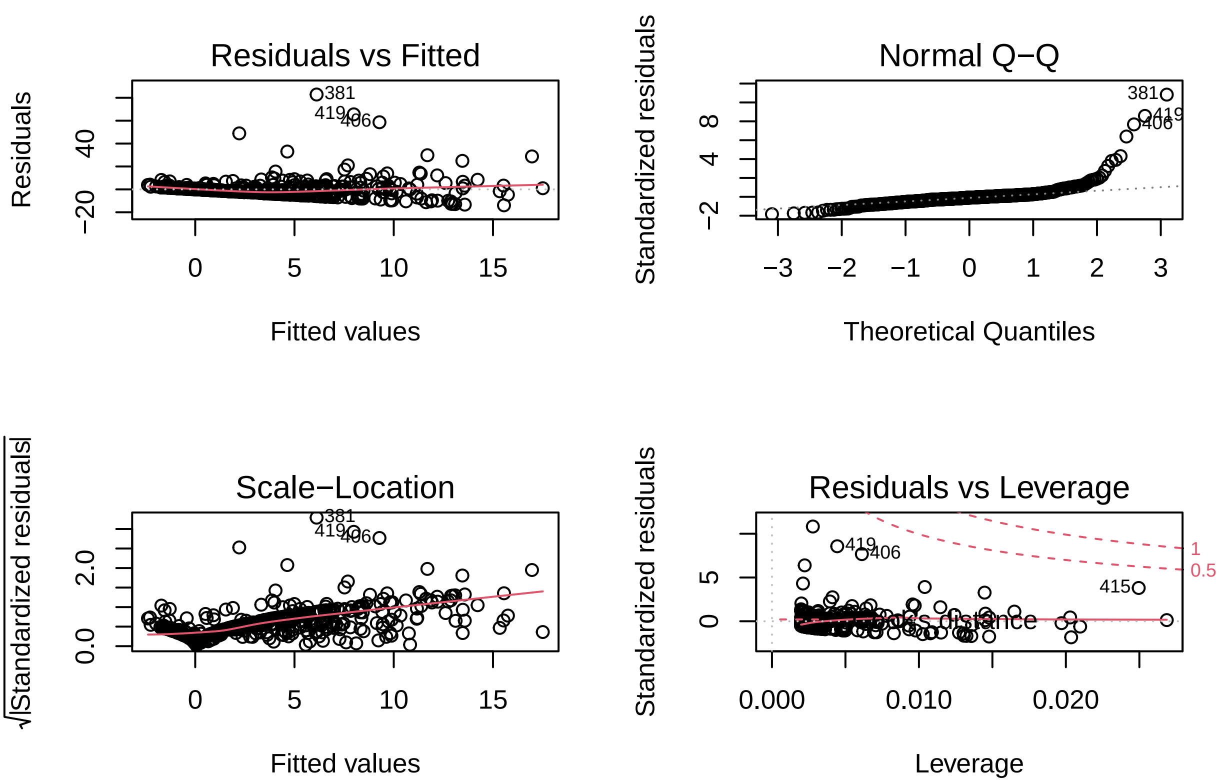
```
## Loading required package: carData
```

```
## The following objects are masked from Boston (pos = 11):
##
##     age, black, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad,
##     rm, tax, zn
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

Crim against lstat

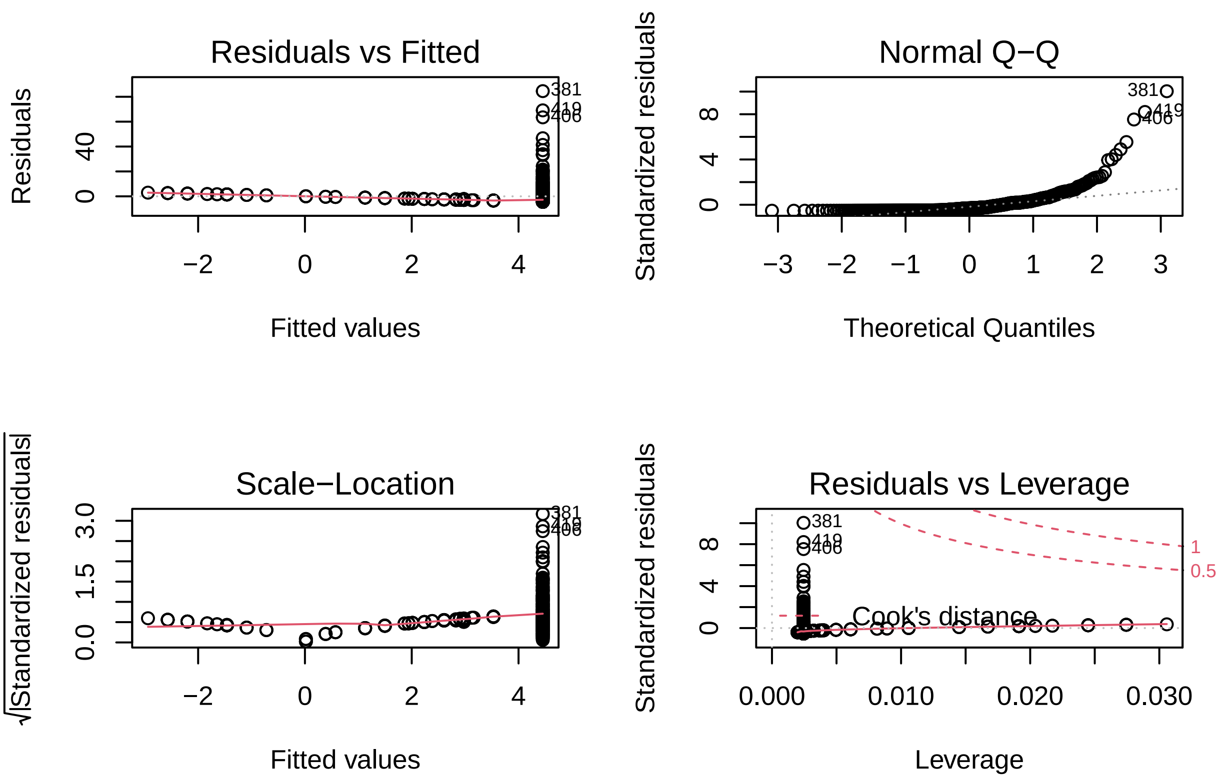
```
##
## Call:
## lm(formula = crim ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```



Crim against zn

```
##
## Call:
```

```
## lm(formula = crim ~ zn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

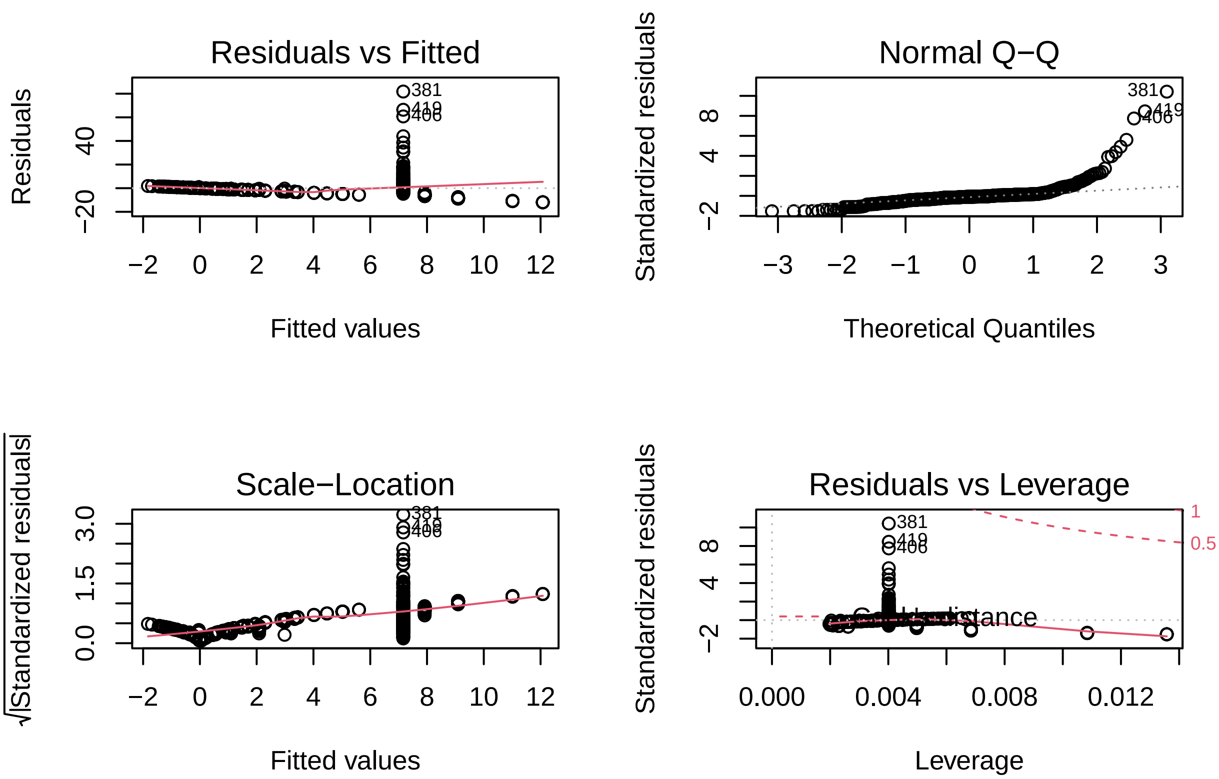


p value = 5.5e-6

Crim against indus

```
##
## Call:
## lm(formula = crim ~ indus)
##
## Residuals:
```

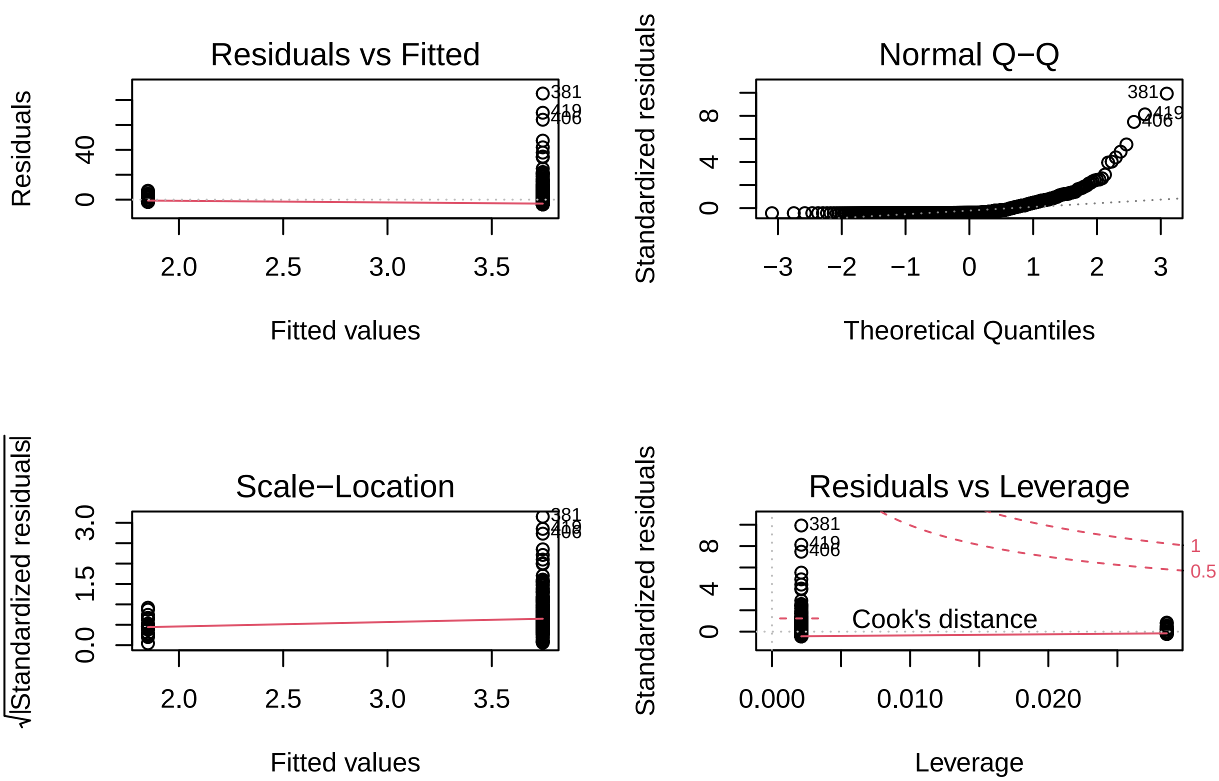
```
##      Min       1Q   Median       3Q      Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
```



Crim against chas

```
##
## Call:
## lm(formula = crim ~ chas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
```

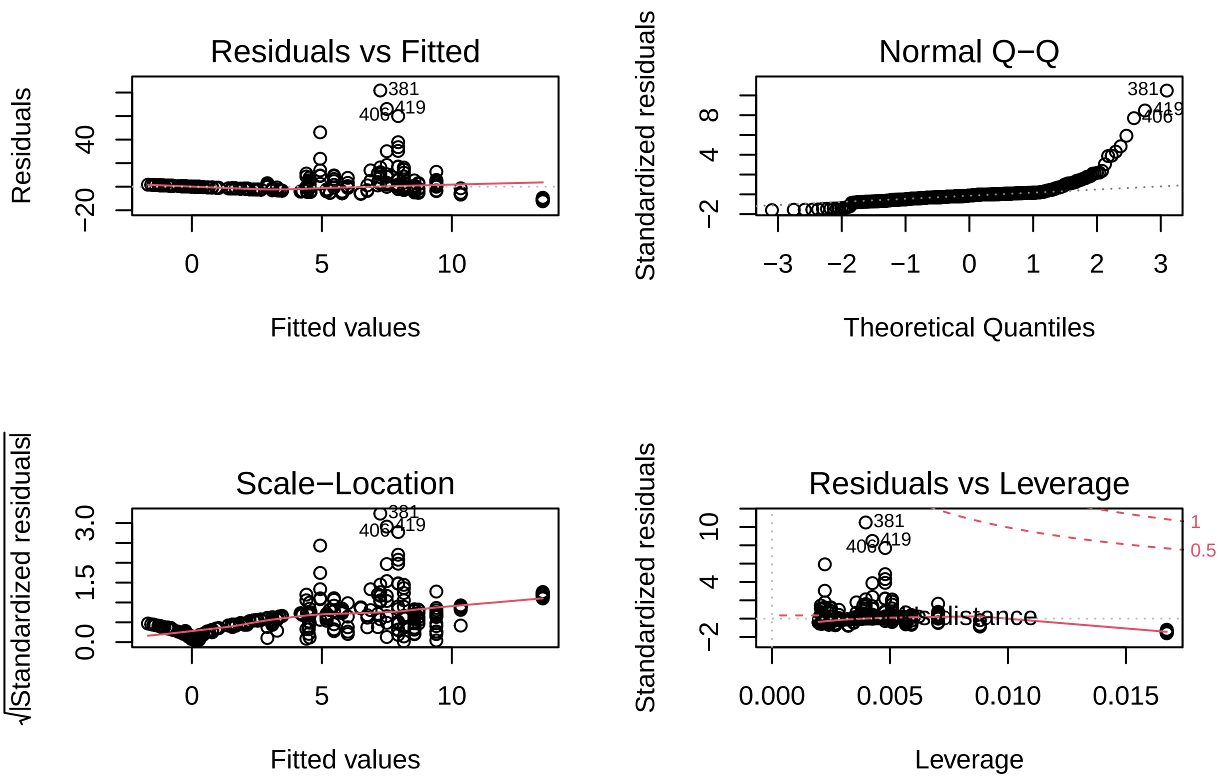
```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas1        -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```



Crim against nox

```
##
## Call:
## lm(formula = crim ~ nox)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419  < 2e-16 ***
## ---
```

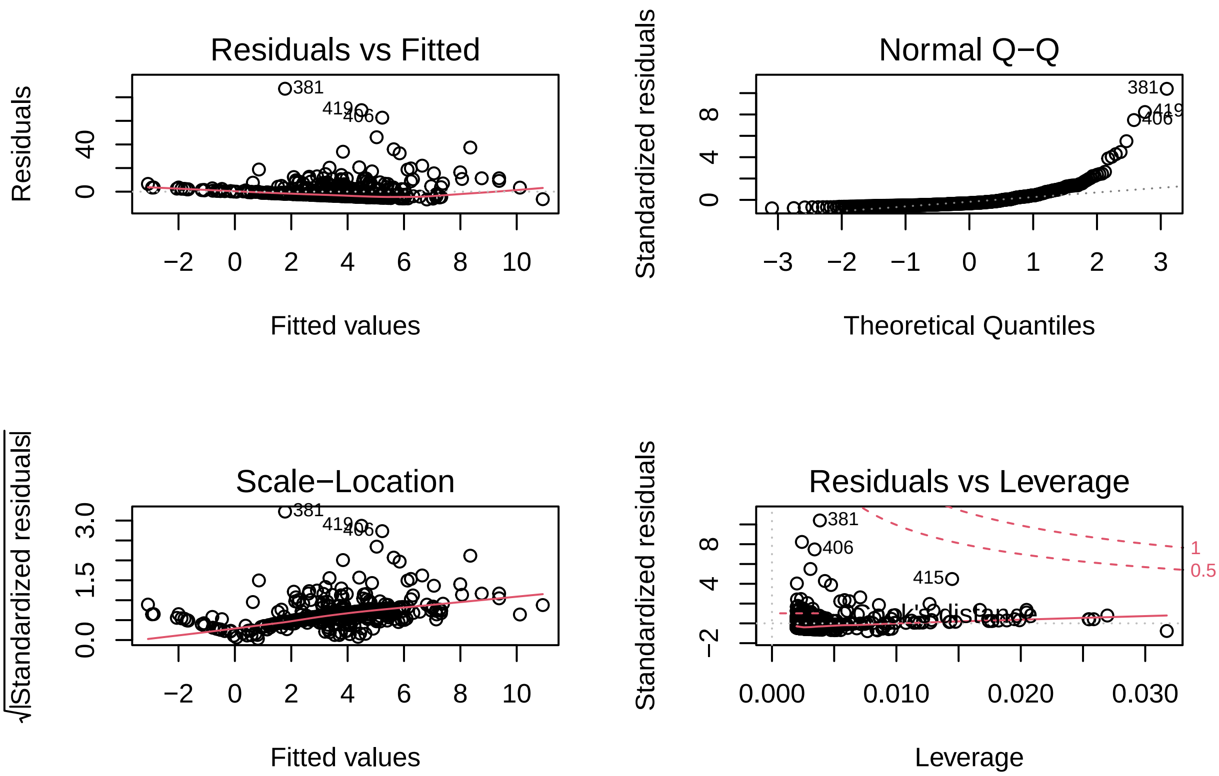
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```



Crim against rm

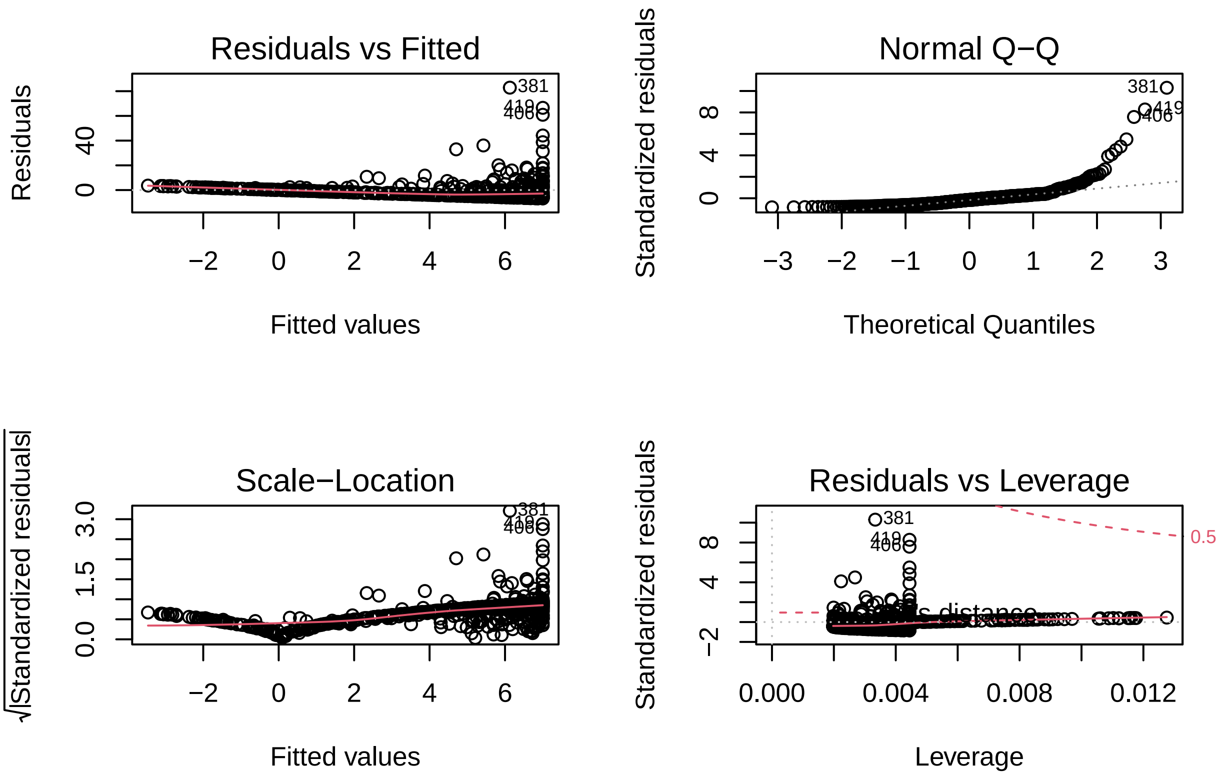
```
##
## Call:
## lm(formula = crim ~ rm)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
```

```
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```



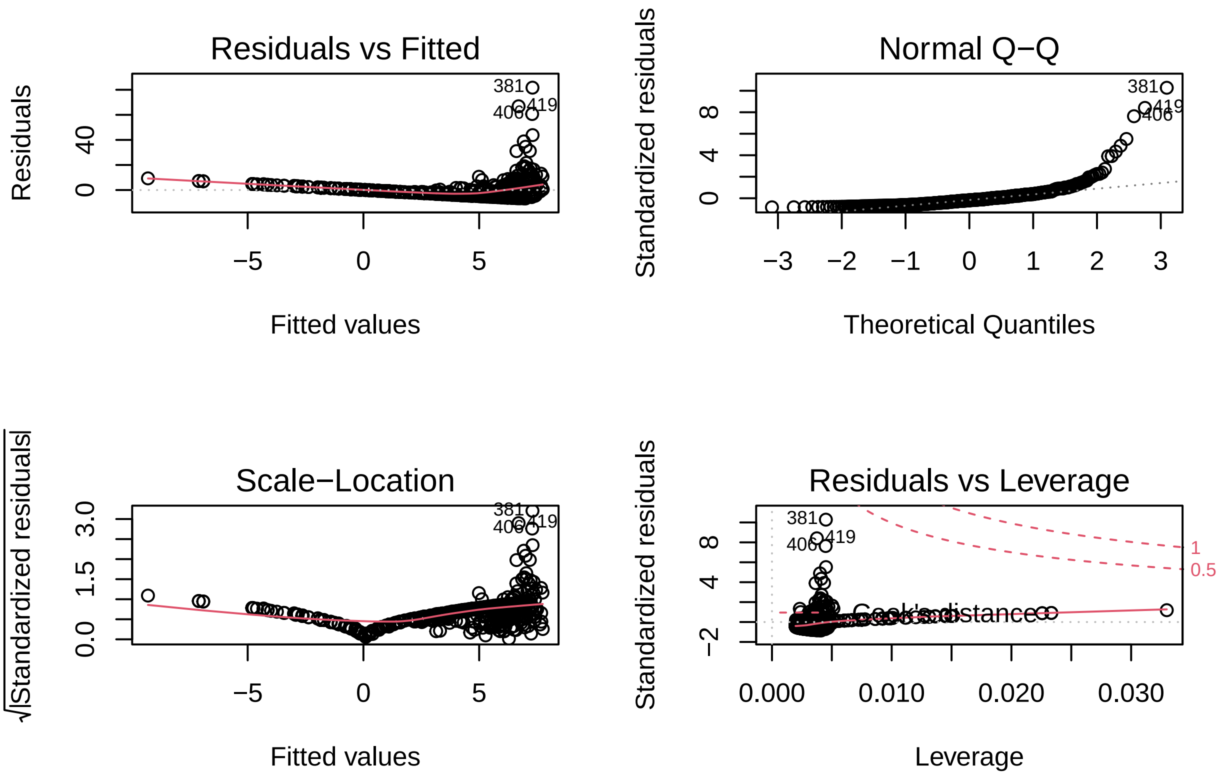
Crim against age

```
##
## Call:
## lm(formula = crim ~ age)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```



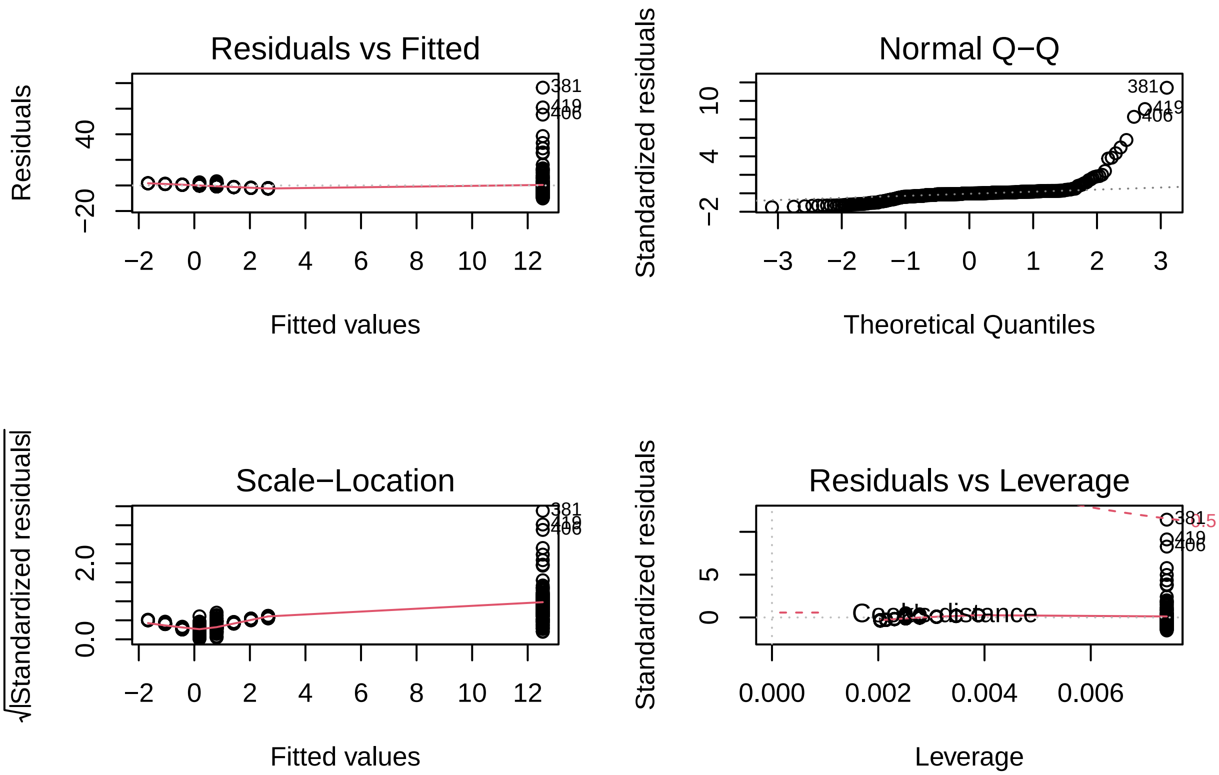
Crim against dis

```
##
## Call:
## lm(formula = crim ~ dis)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```



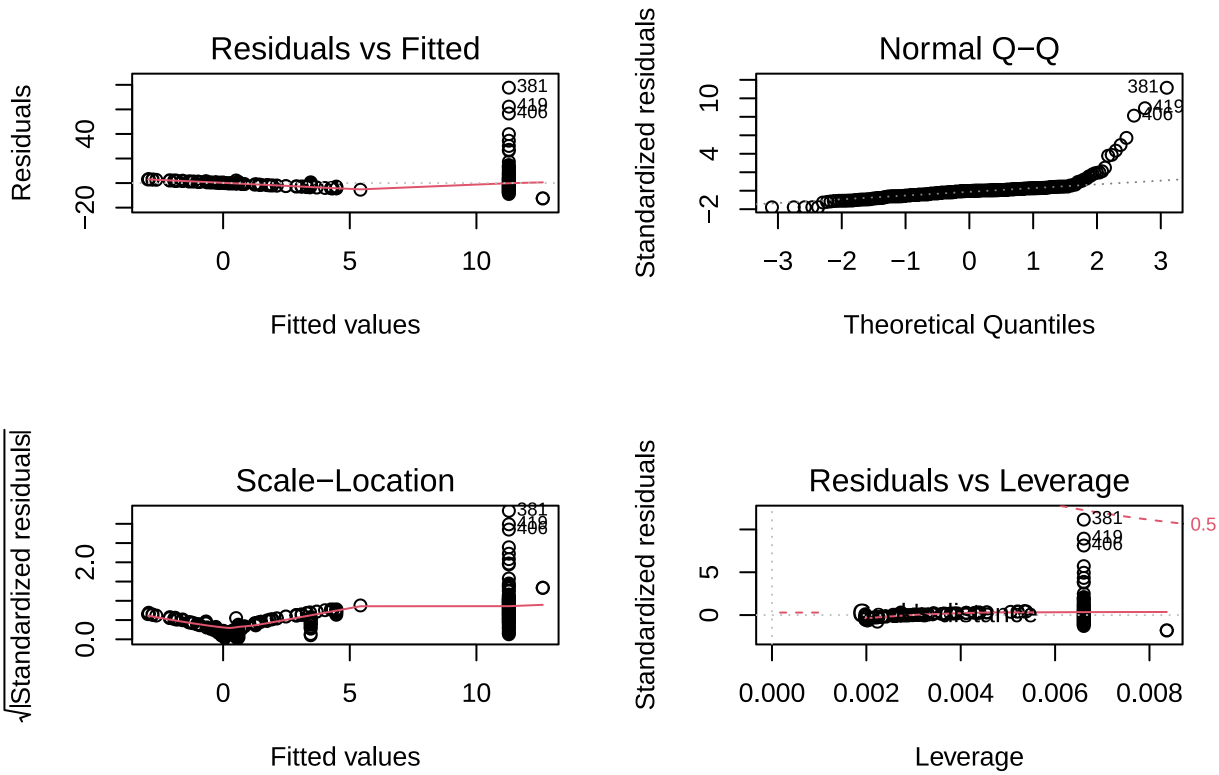
Crim against rad

```
##
## Call:
## lm(formula = crim ~ rad)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```



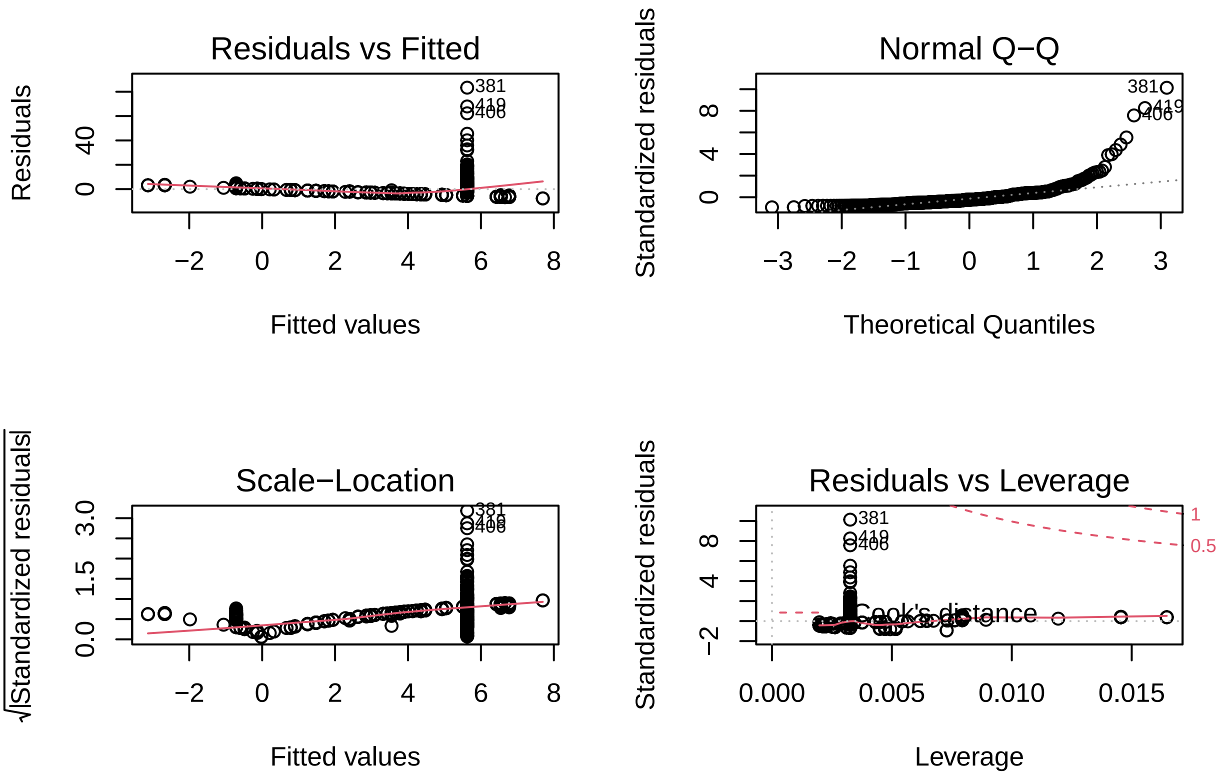
Crim against tax

```
##
## Call:
## lm(formula = crim ~ tax)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```



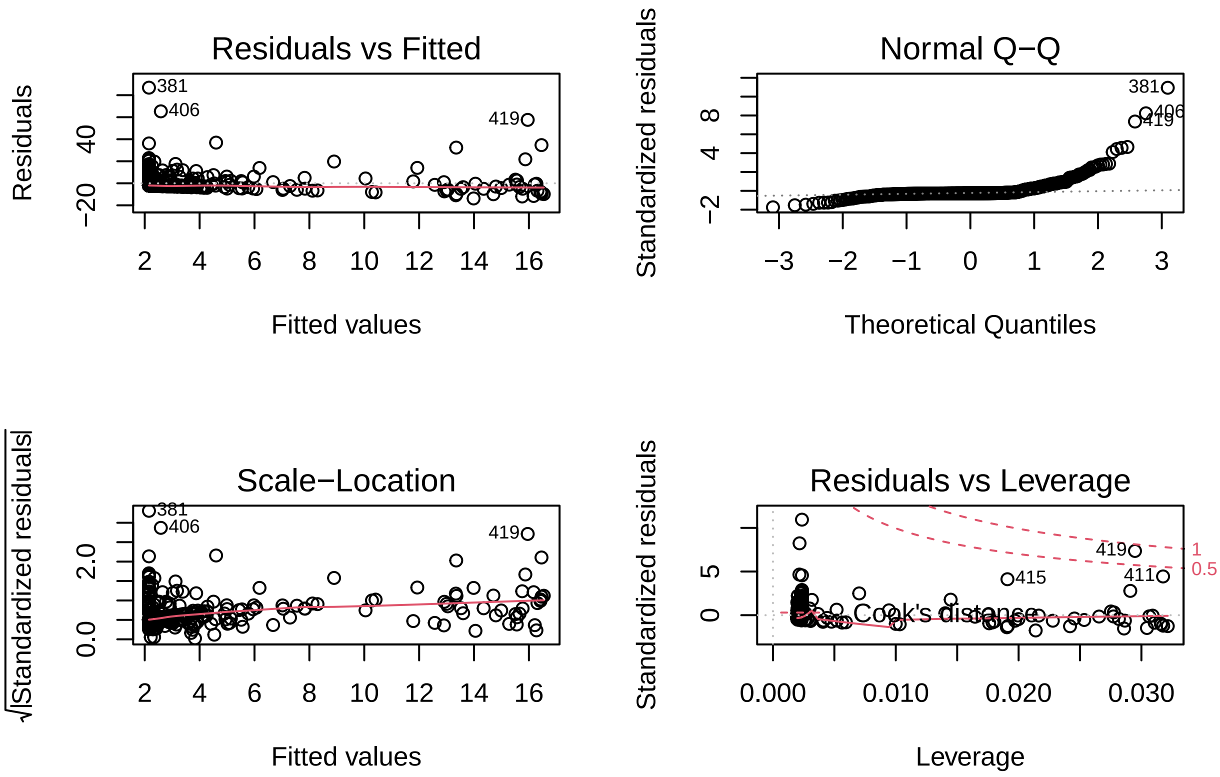
Crim against ptratio

```
##
## Call:
## lm(formula = crim ~ ptratio)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```



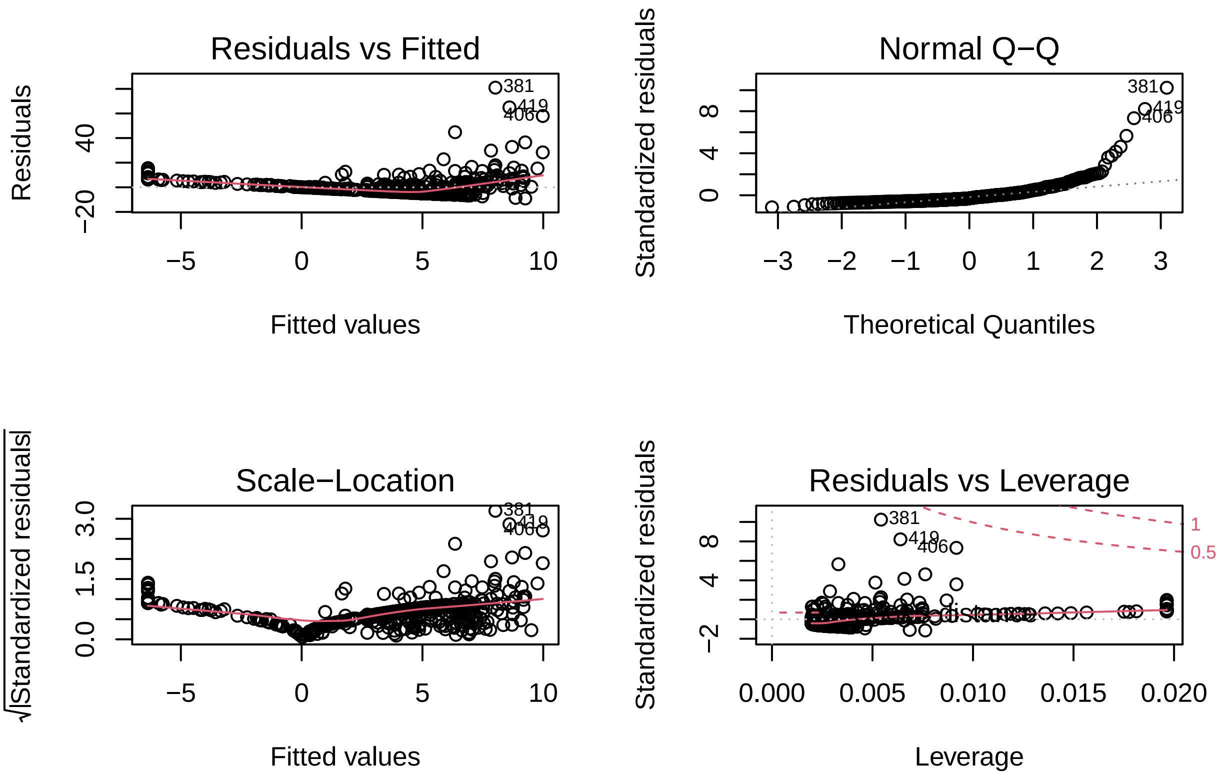
Crim against black

```
##
## Call:
## lm(formula = crim ~ black)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -0.036280   0.003873  -9.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```



Crim against medv

```
##
## Call:
## lm(formula = crim ~ medv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

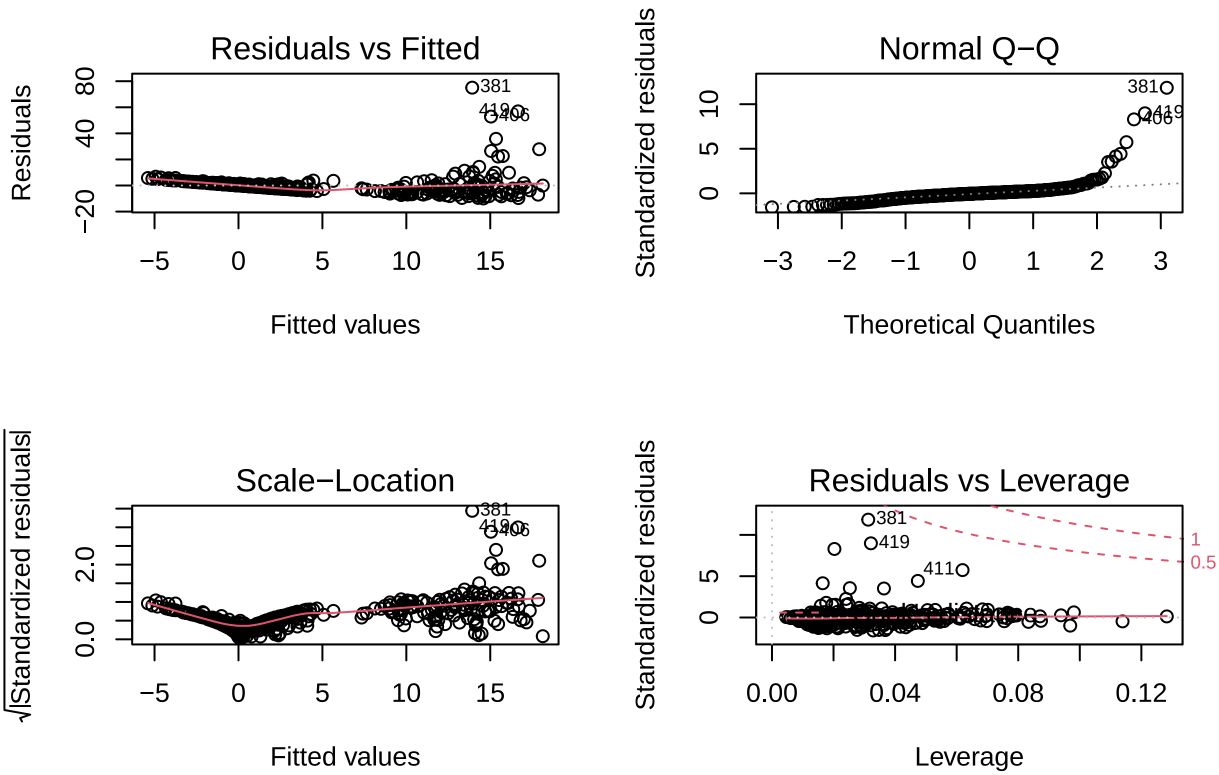


1.Lstat p value < 2.2e-16, Low standard error, high impact variable
i.e statistically significant. Lower income neighborhoods can tend to have higher crime rates
2.Industry p value < 2.2e-16, high impact variable i.e statistically significant.
The higher the non retail business, the higher the crime i.e large city
3.Proximity to Charles River (Refactoring boolean variable) p value = 0.2094
Proximity to Charles river is not indicative of crime rate
4.Nox level value < 2.2e-16, high impact variable i.e statistically significant
Nox levels may indicate health problems and poverty, driving up crime
5.Room count p value = 6.347e-7
Higher room neighborhoods have lower crime due to being upscale
6.Age p value = 2.2855e-16, high impact variable i.e statistically significant
Older neighbourhoods can have higher crime rates due to higher population
7.dis p value < 2.2e-16, high impact variable i.e statistically significant
As the distance to employment increases, crime might increase due to lack of employment
8.rad p value < 2.2e-16, high impact variable i.e statistically significant
Higher crime rates around radial highways due to illegal activity under said highways
9.Tax value p value < 2.2e-16, high impact variable i.e statistically significant
Higher tax rates indicate wealthier neighborhoods with lower crime
10.PT Ratio p value=2.943e-11
Higher Pupil teacher ration increases crime due to reduced access and effectiveness of education
11.black p value < 2.2e-16
High impact variable i.e statistically significant
12.Median housing value p value < 2.2e-16,

High impact variable i.e statistically significant
Lower median house value in poorer neighborhoods, higher crime

Part B

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

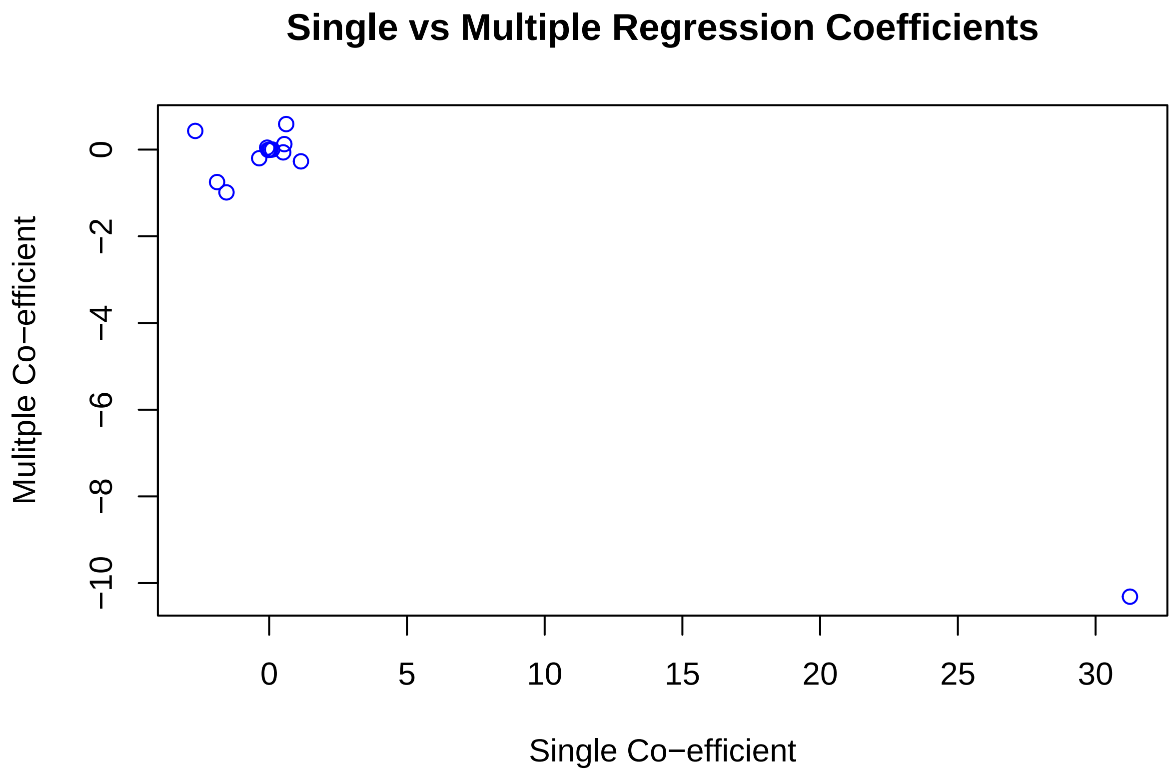



```
##       zn    indus     chas      nox       rm      age      dis      rad
## 2.325094 3.987753 1.094326 4.551563 2.258113 3.100801 4.289041 7.158834
##      tax  ptratio    black    lstat     medv
## 9.195495 1.984489 1.369741 3.561476 3.772856
```

```
##                      2.5 %        97.5 %
## (Intercept)    2.818109179 31.2483458660
## zn             0.008046562  0.0816638671
## indus         -0.227733150  0.1000235023
## chas          -3.067882868  1.5696156471
## nox          -20.678894713  0.0518248891
## rm            -0.773956866  1.6342178774
## age           -0.033767600  0.0366708869
## dis           -1.540889544 -0.4334619069
## rad            0.415209611  0.7612075719
## tax           -0.013909700  0.0063496670
## ptratio       -0.637417996  0.0952568794
## black         -0.014754837 -0.0003201725
## lstat         -0.022572584  0.2749953365
## medv          -0.317788478 -0.0799851646
```

We can reject the null hypothesis for zn, dis, rad, black, and medv at the 5% confidence interval.

Part C

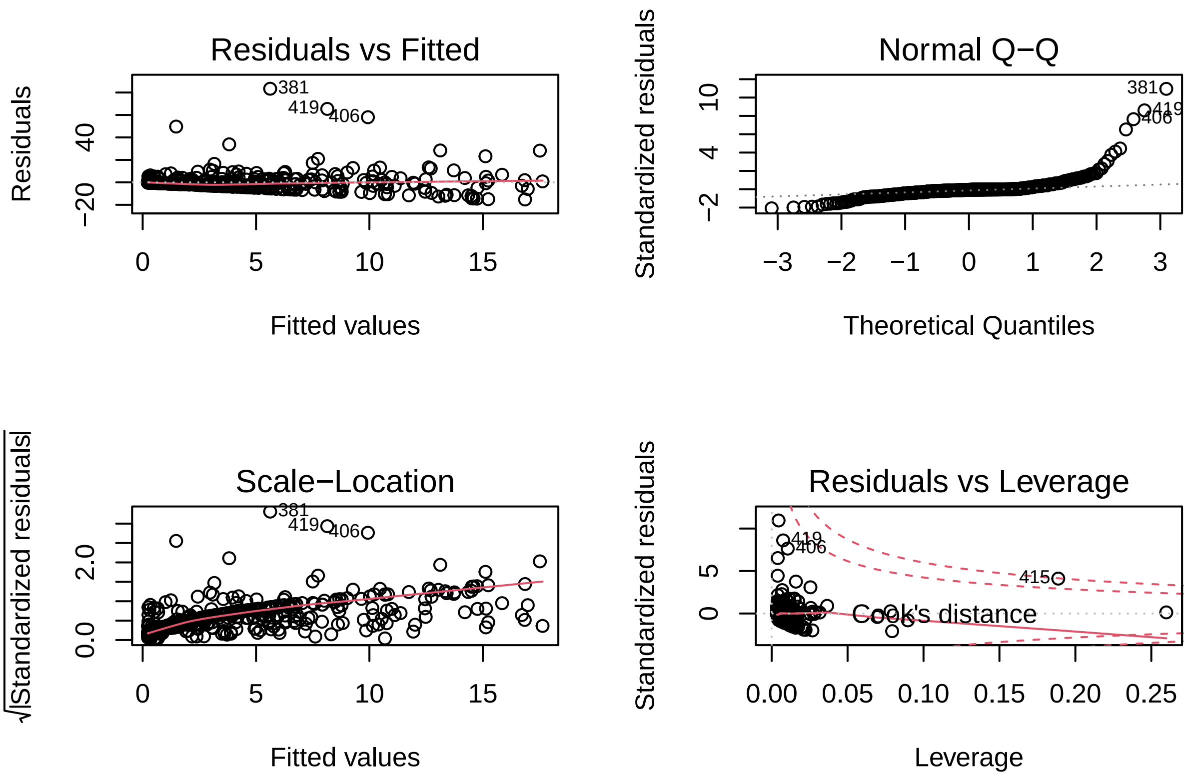


Part D

Crim against lstat

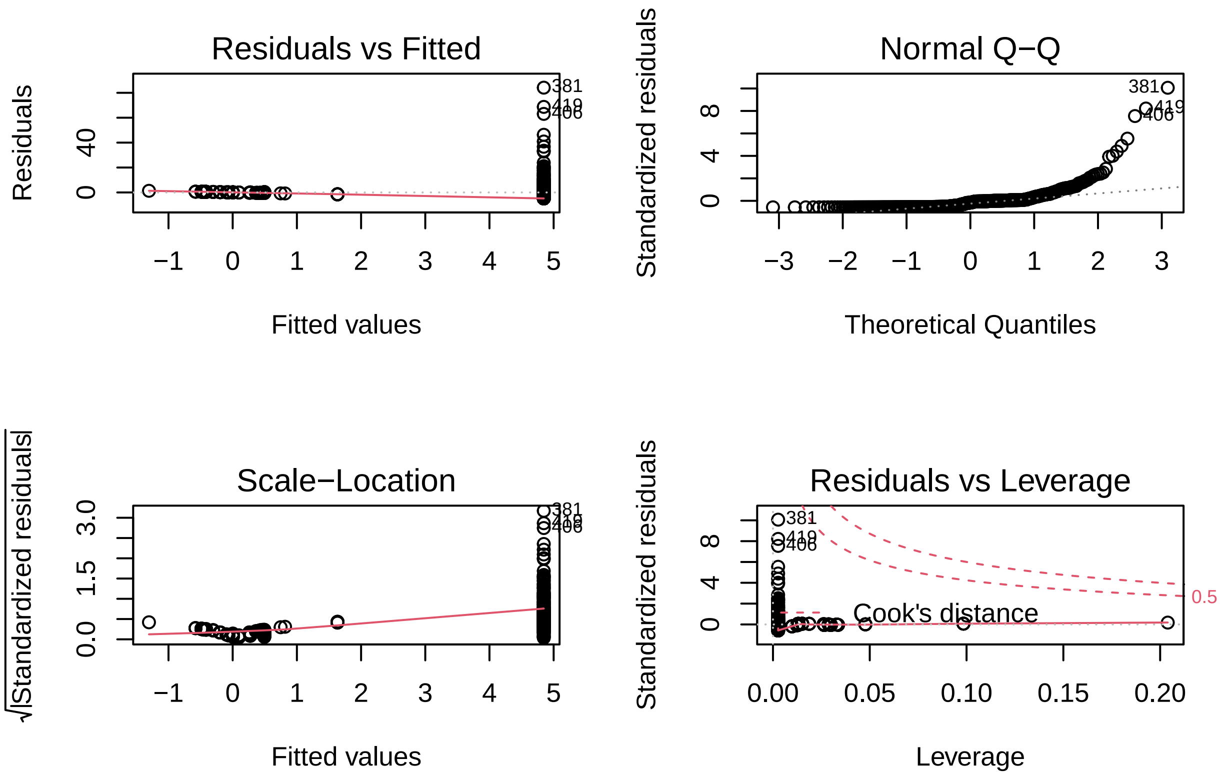
```
##
## Call:
## lm(formula = crim ~ poly(lstat, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3392  10.654   <2e-16 ***
## poly(lstat, 3)1  88.0697     7.6294  11.543   <2e-16 ***
## poly(lstat, 3)2  15.8882     7.6294   2.082   0.0378 *
## poly(lstat, 3)3 -11.5740     7.6294  -1.517   0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
```

```
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
```



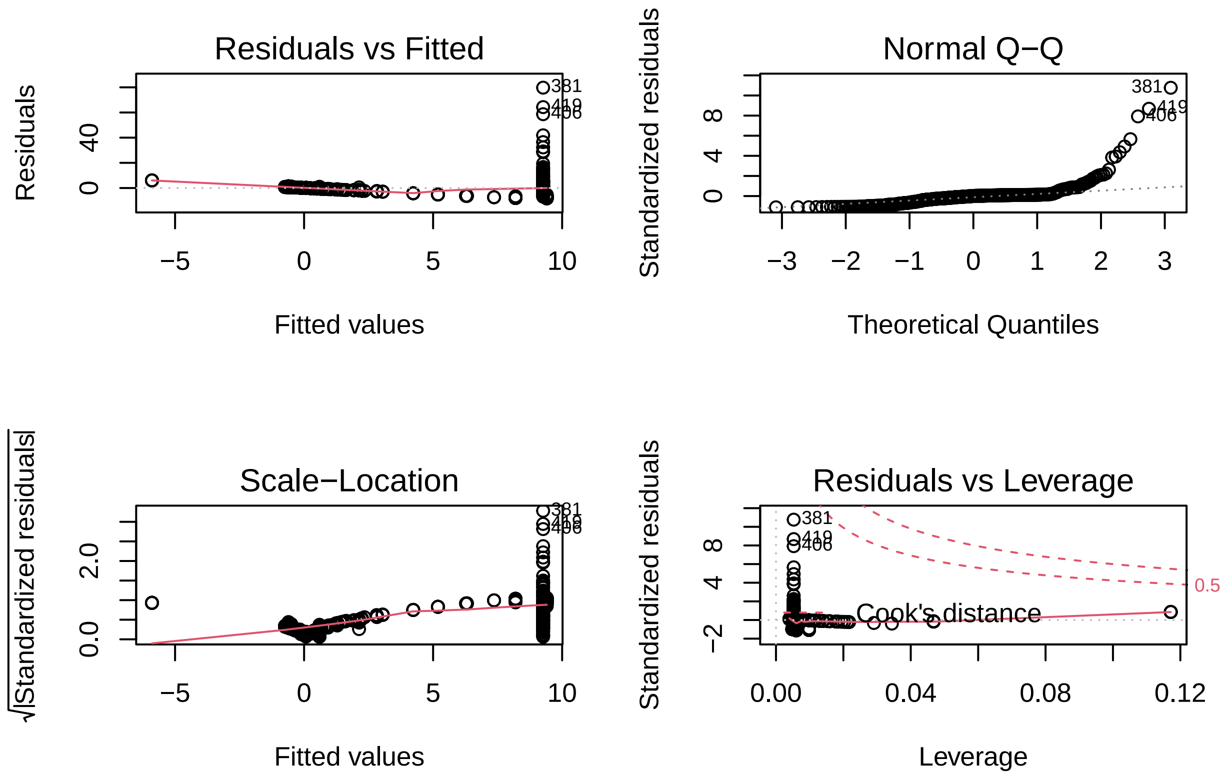
Crim against zn

```
##
## Call:
## lm(formula = crim ~ poly(zn, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3722   9.709  < 2e-16 ***
## poly(zn, 3)1 -38.7498     8.3722  -4.628  4.7e-06 ***
## poly(zn, 3)2  23.9398     8.3722   2.859  0.00442 **
## poly(zn, 3)3 -10.0719     8.3722  -1.203  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
```



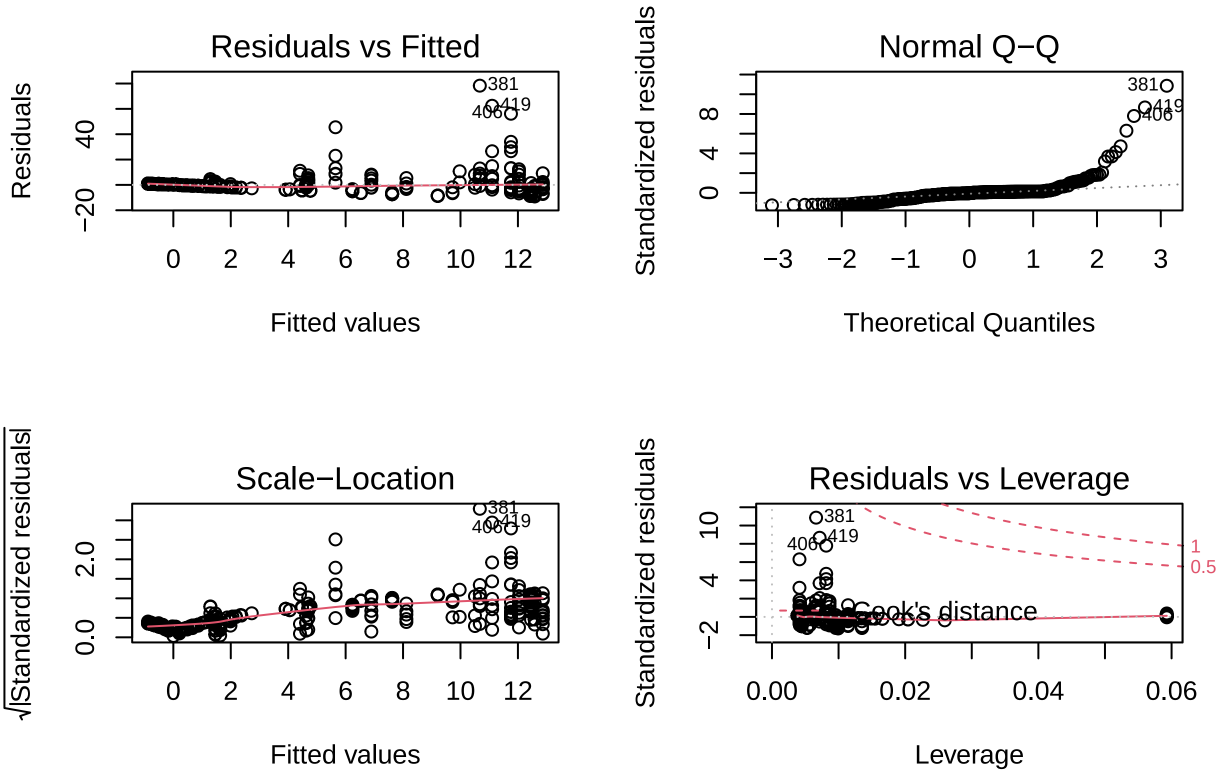
Crim against indus

```
##
## Call:
## lm(formula = crim ~ poly(indus, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.614      0.330  10.950  < 2e-16 ***
## poly(indus, 3)1   78.591      7.423  10.587  < 2e-16 ***
## poly(indus, 3)2  -24.395      7.423  -3.286  0.00109 **
## poly(indus, 3)3  -54.130      7.423  -7.292  1.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16
```



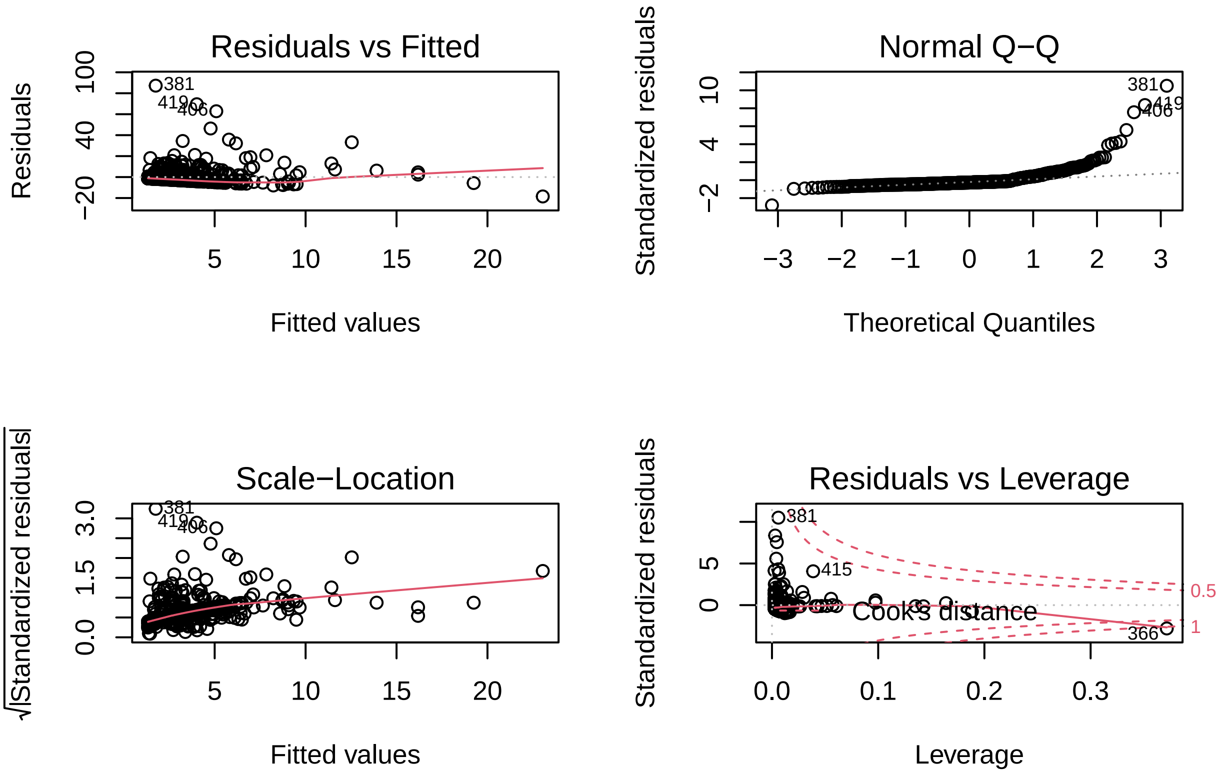
Crim against nox

```
##
## Call:
## lm(formula = crim ~ poly(nox, 3))
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -9.110 -2.068 -0.255  0.739 78.302
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3216  11.237  < 2e-16 ***
## poly(nox, 3)1  81.3720     7.2336  11.249  < 2e-16 ***
## poly(nox, 3)2 -28.8286     7.2336  -3.985 7.74e-05 ***
## poly(nox, 3)3 -60.3619     7.2336  -8.345 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16
```



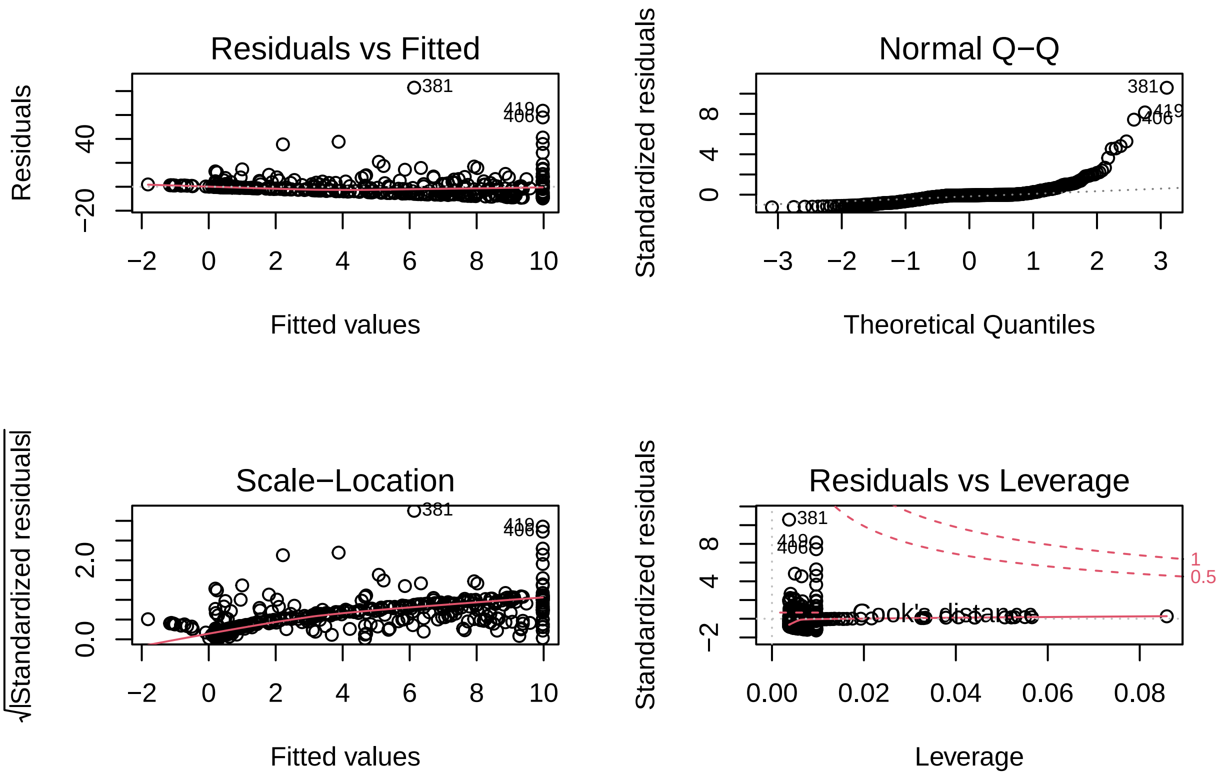
Crim against rm

```
##
## Call:
## lm(formula = crim ~ poly(rm, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.485  -3.468  -2.221  -0.015  87.219
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3703   9.758  < 2e-16 ***
## poly(rm, 3)1 -42.3794     8.3297  -5.088 5.13e-07 ***
## poly(rm, 3)2  26.5768     8.3297   3.191  0.00151 **
## poly(rm, 3)3  -5.5103     8.3297  -0.662  0.50858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,    Adjusted R-squared:  0.06222
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
```



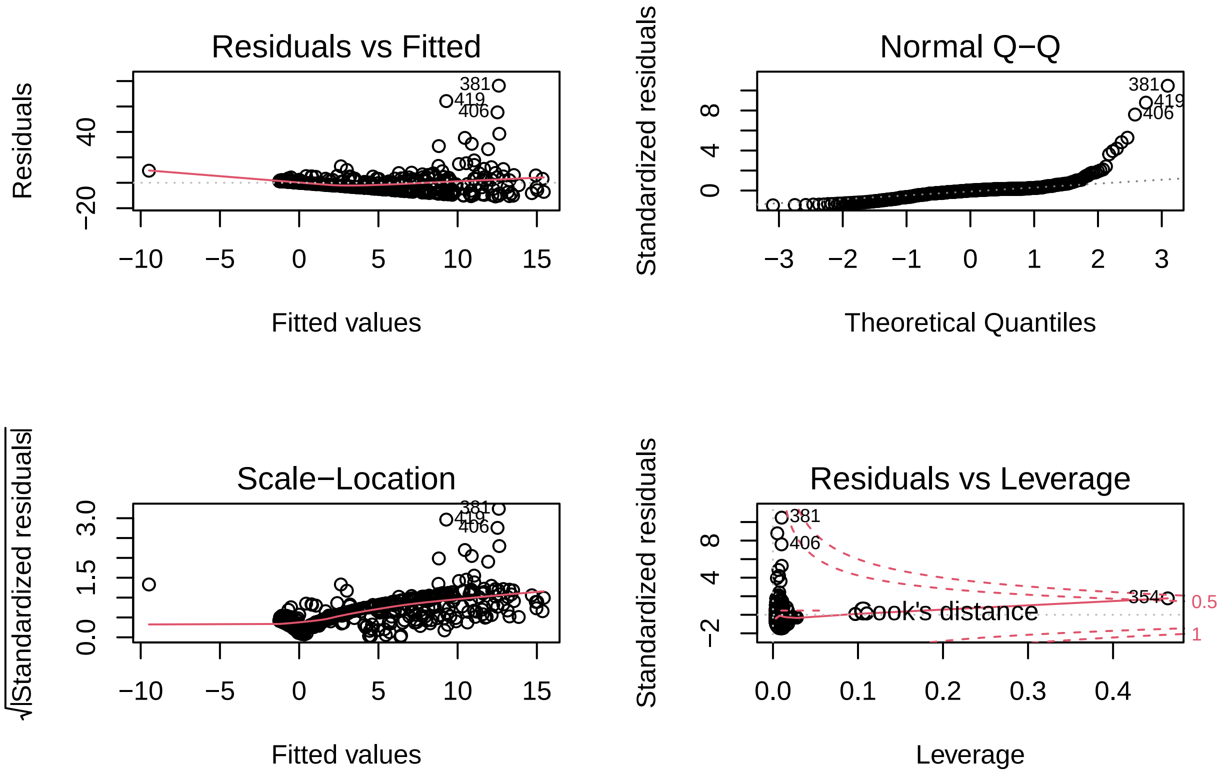
Crim against age

```
##
## Call:
## lm(formula = crim ~ poly(age, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.762 -2.673 -0.516  0.019 82.842
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3485  10.368  < 2e-16 ***
## poly(age, 3)1  68.1820     7.8397   8.697  < 2e-16 ***
## poly(age, 3)2  37.4845     7.8397   4.781 2.29e-06 ***
## poly(age, 3)3  21.3532     7.8397   2.724  0.00668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16
```



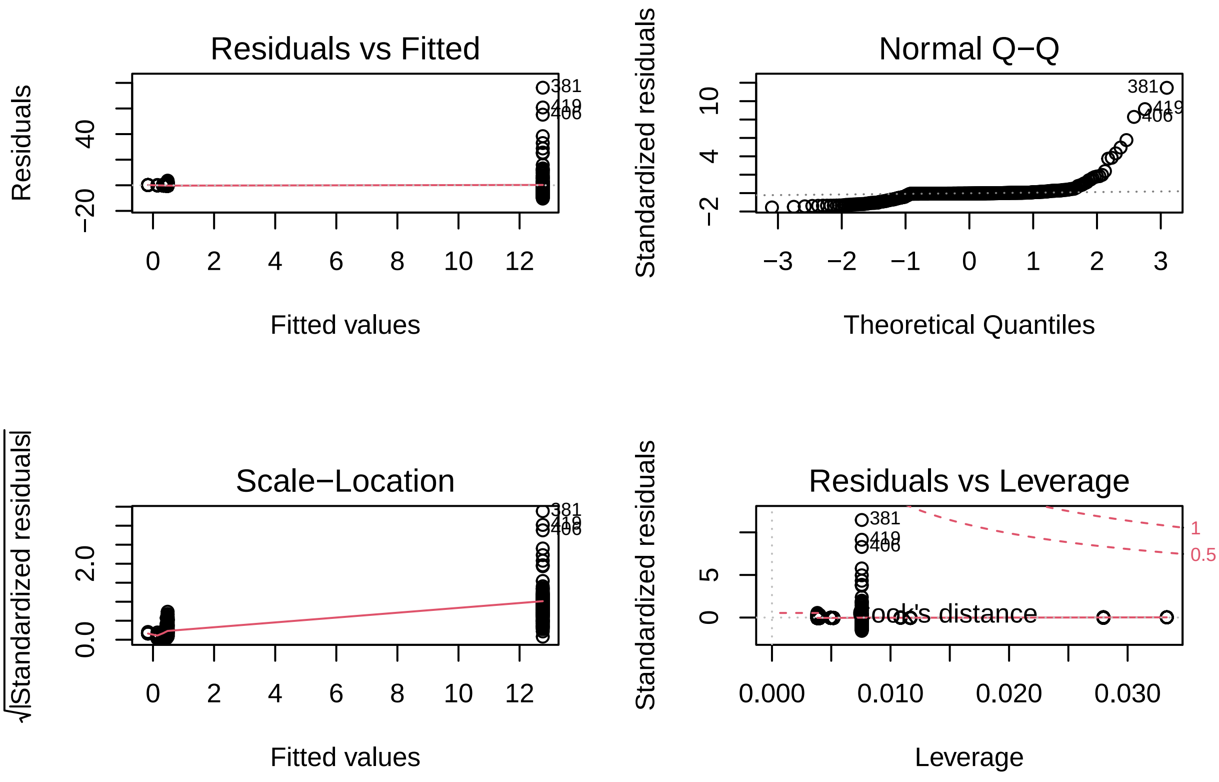
Crim against dis

```
##
## Call:
## lm(formula = crim ~ poly(dis, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3259  11.087  < 2e-16 ***
## poly(dis, 3)1 -73.3886     7.3315 -10.010  < 2e-16 ***
## poly(dis, 3)2  56.3730     7.3315   7.689 7.87e-14 ***
## poly(dis, 3)3 -42.6219     7.3315  -5.814 1.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
```



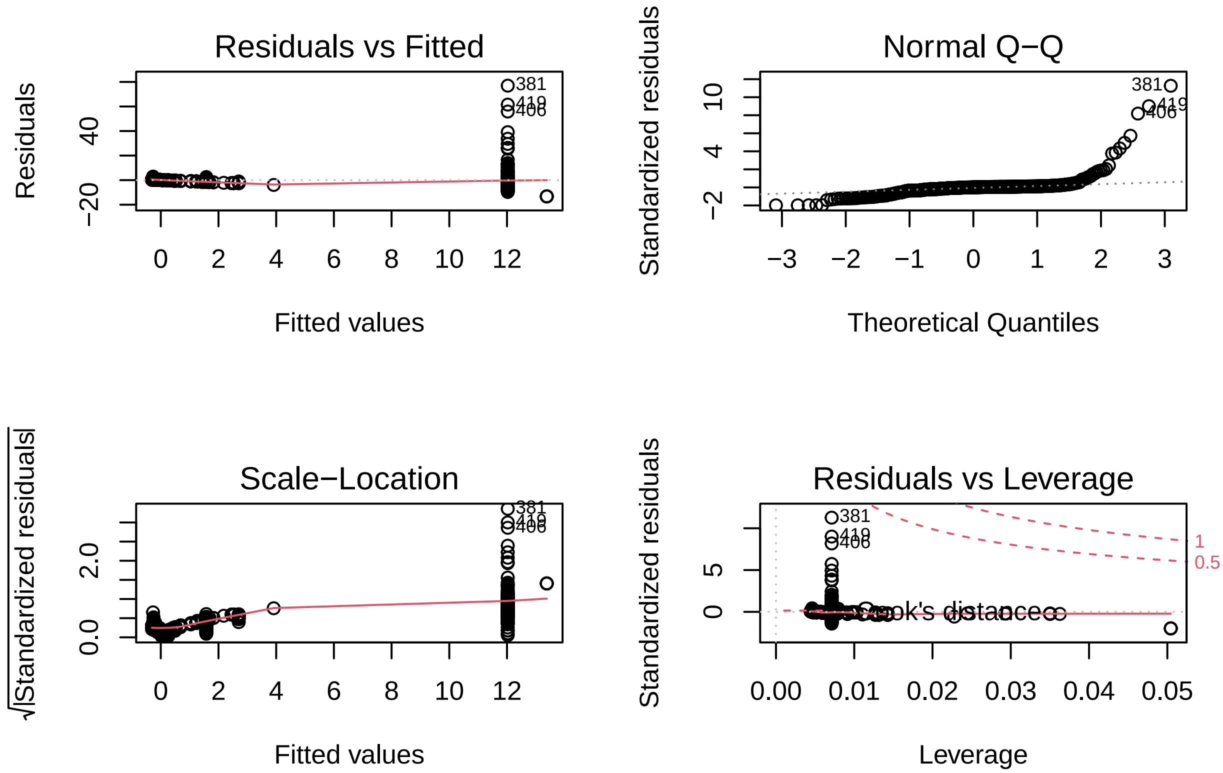
Crim against rad

```
##
## Call:
## lm(formula = crim ~ poly(rad, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.6135     0.2971  12.164  < 2e-16 ***
## poly(rad, 3)1 120.9074     6.6824  18.093  < 2e-16 ***
## poly(rad, 3)2  17.4923     6.6824   2.618  0.00912 **
## poly(rad, 3)3   4.6985     6.6824   0.703  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
```



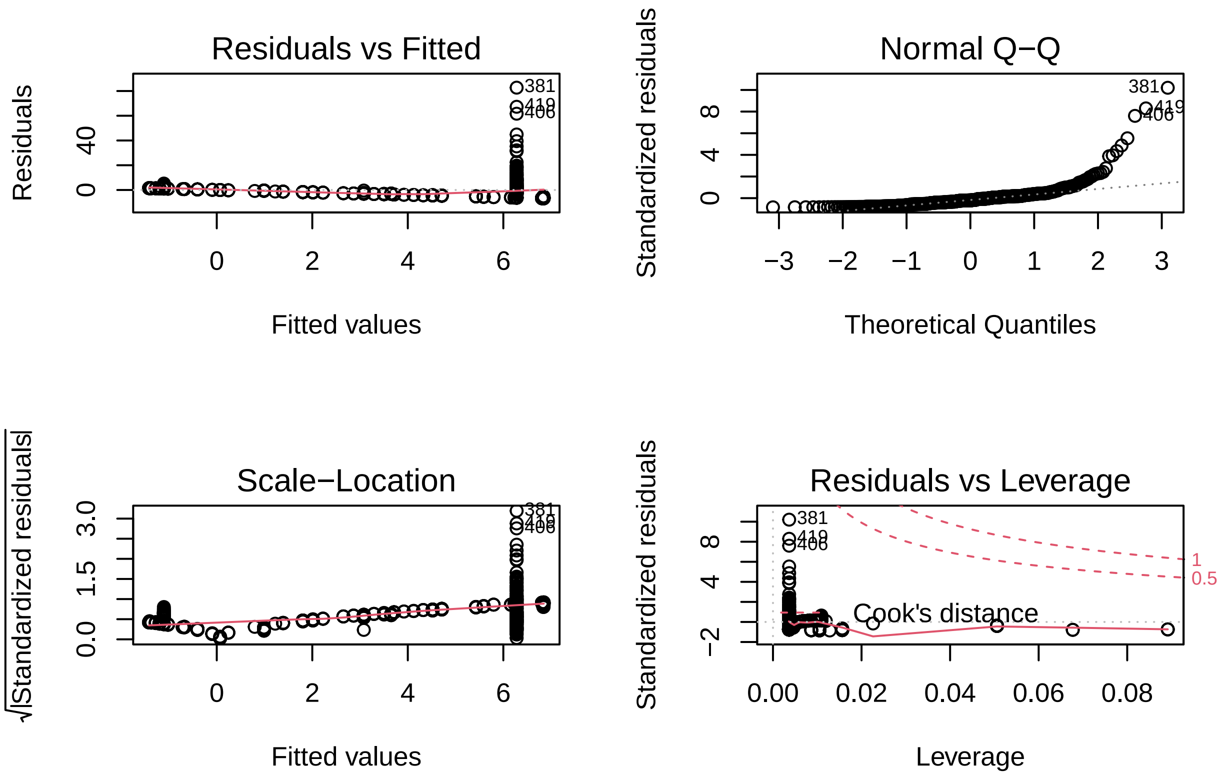
Crim against tax

```
##
## Call:
## lm(formula = crim ~ poly(tax, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.6135     0.3047  11.860  < 2e-16 ***
## poly(tax, 3)1 112.6458     6.8537  16.436  < 2e-16 ***
## poly(tax, 3)2  32.0873     6.8537   4.682 3.67e-06 ***
## poly(tax, 3)3  -7.9968     6.8537  -1.167    0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.854 on 502 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16
```



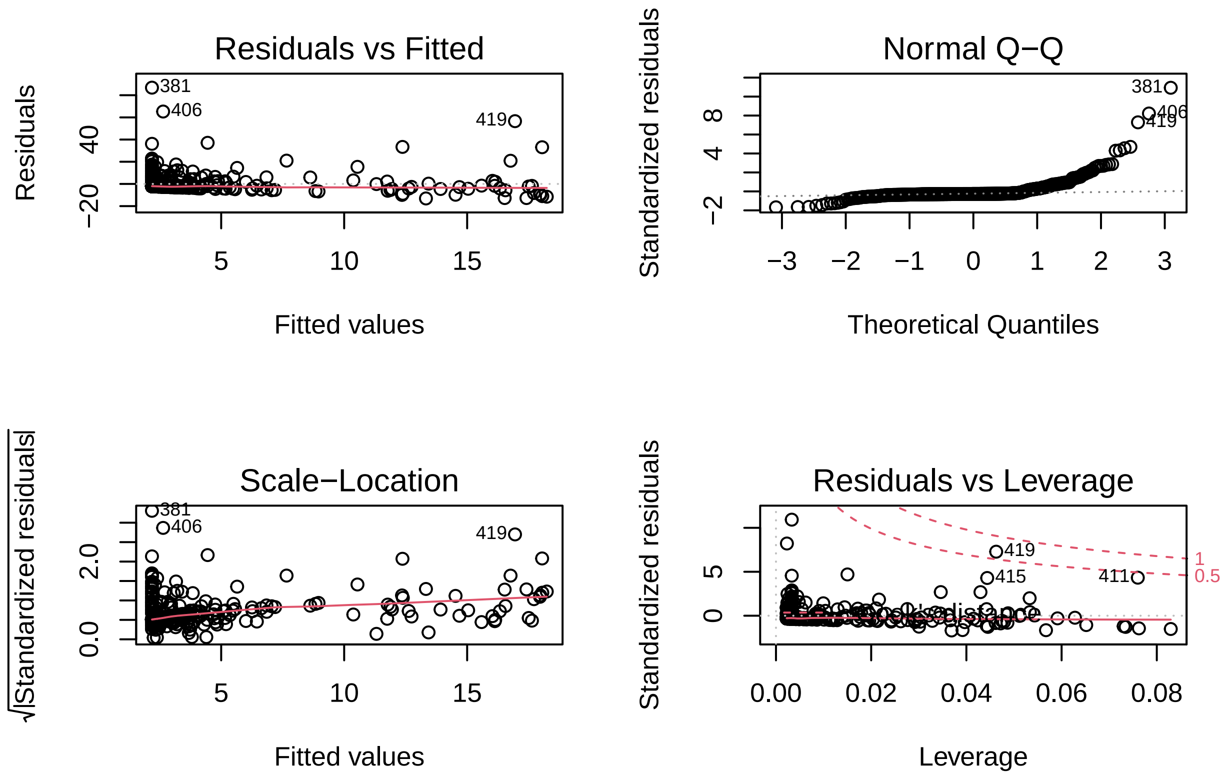
Crim against ptratio

```
##
## Call:
## lm(formula = crim ~ poly(ptratio, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.614      0.361  10.008  < 2e-16 ***
## poly(ptratio, 3)1   56.045      8.122   6.901 1.57e-11 ***
## poly(ptratio, 3)2   24.775      8.122   3.050  0.00241 **
## poly(ptratio, 3)3  -22.280      8.122  -2.743  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
```



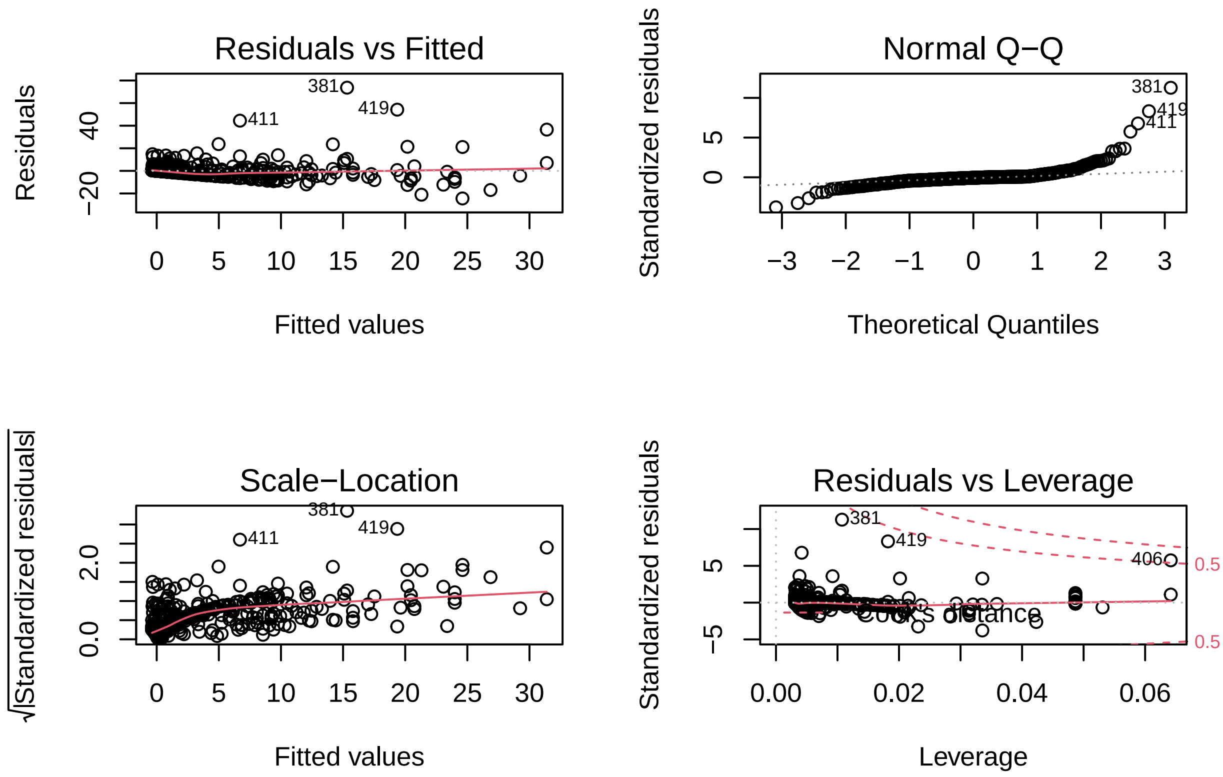
Crim against black

```
##
## Call:
## lm(formula = crim ~ poly(black, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.096  -2.343  -2.128  -1.439  86.790
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3536  10.218   <2e-16 ***
## poly(black, 3)1 -74.4312     7.9546  -9.357   <2e-16 ***
## poly(black, 3)2   5.9264     7.9546   0.745    0.457
## poly(black, 3)3  -4.8346     7.9546  -0.608    0.544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.955 on 502 degrees of freedom
## Multiple R-squared:  0.1498, Adjusted R-squared:  0.1448
## F-statistic: 29.49 on 3 and 502 DF,  p-value: < 2.2e-16
```



Crim against medv

```
##
## Call:
## lm(formula = crim ~ poly(medv, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.614      0.292  12.374  < 2e-16 ***
## poly(medv, 3)1  -75.058      6.569 -11.426  < 2e-16 ***
## poly(medv, 3)2   88.086      6.569  13.409  < 2e-16 ***
## poly(medv, 3)3  -48.033      6.569  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

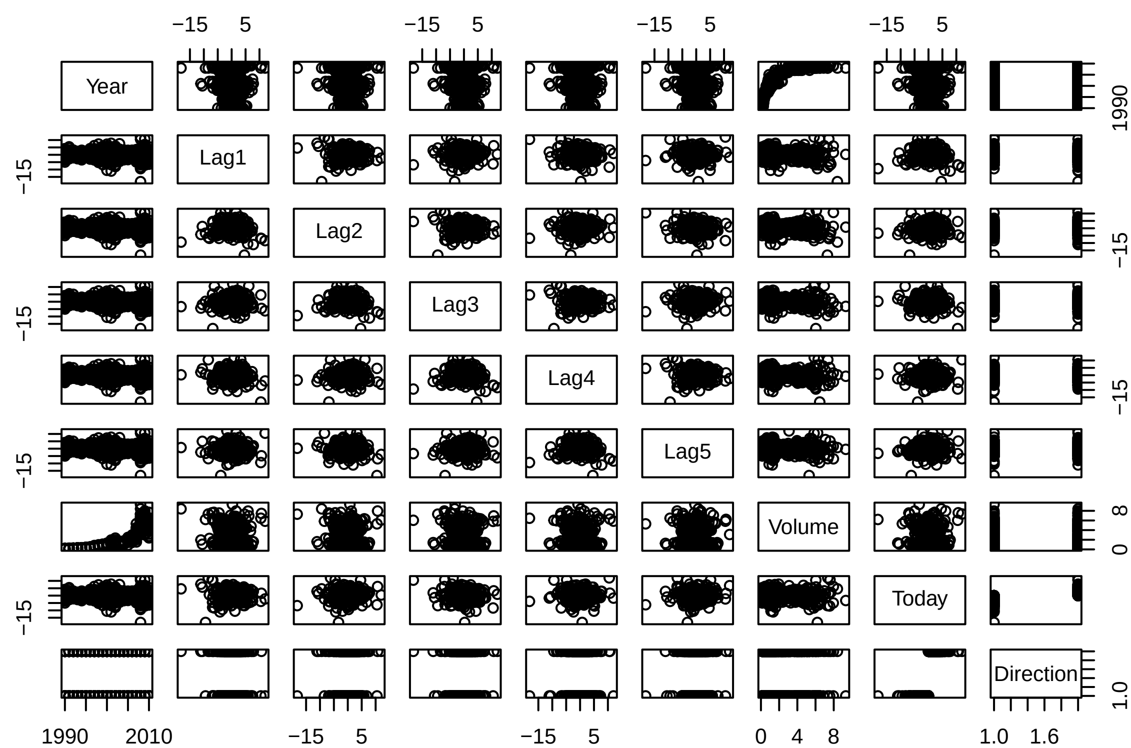


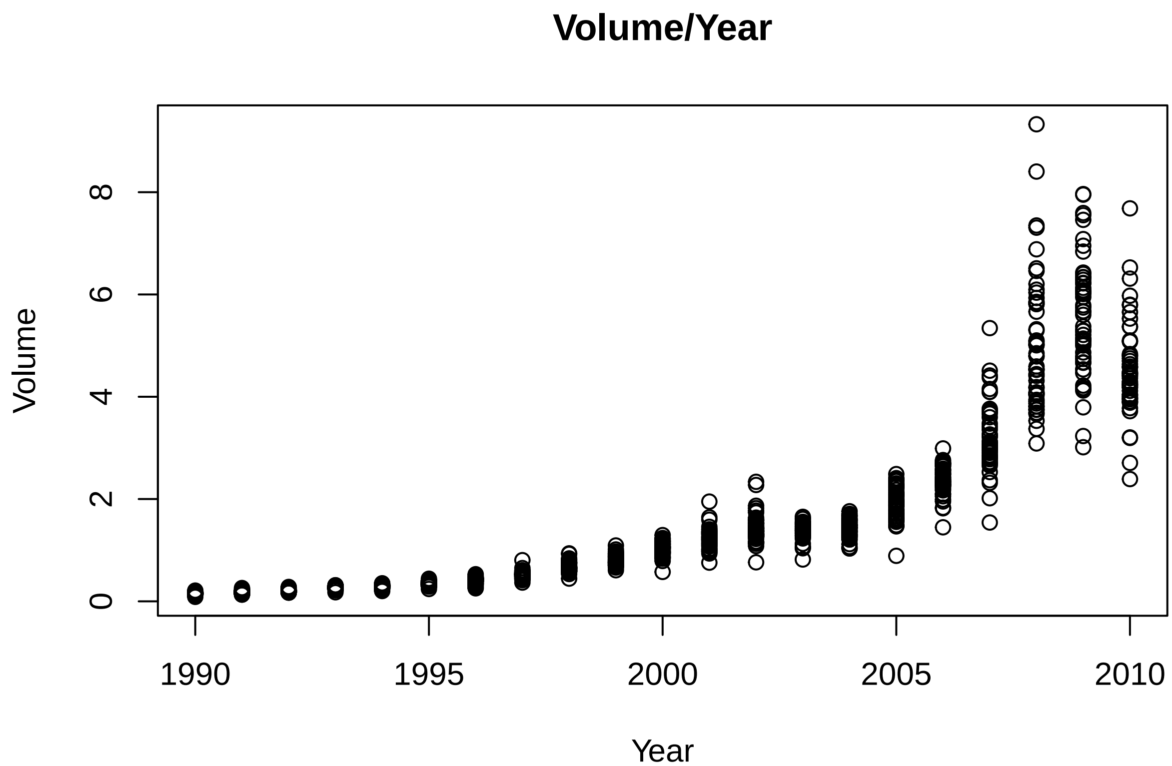
Chapter 4 Question 10

Part A

```
##       Year           Lag1                Lag2                Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4                Lag5              Volume            Today
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
##  Direction
##  Down:484
##  Up  :605
##
##
##
```

```
##

## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"

##                Year         Lag1        Lag2        Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##                Lag5      Volume        Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000
```



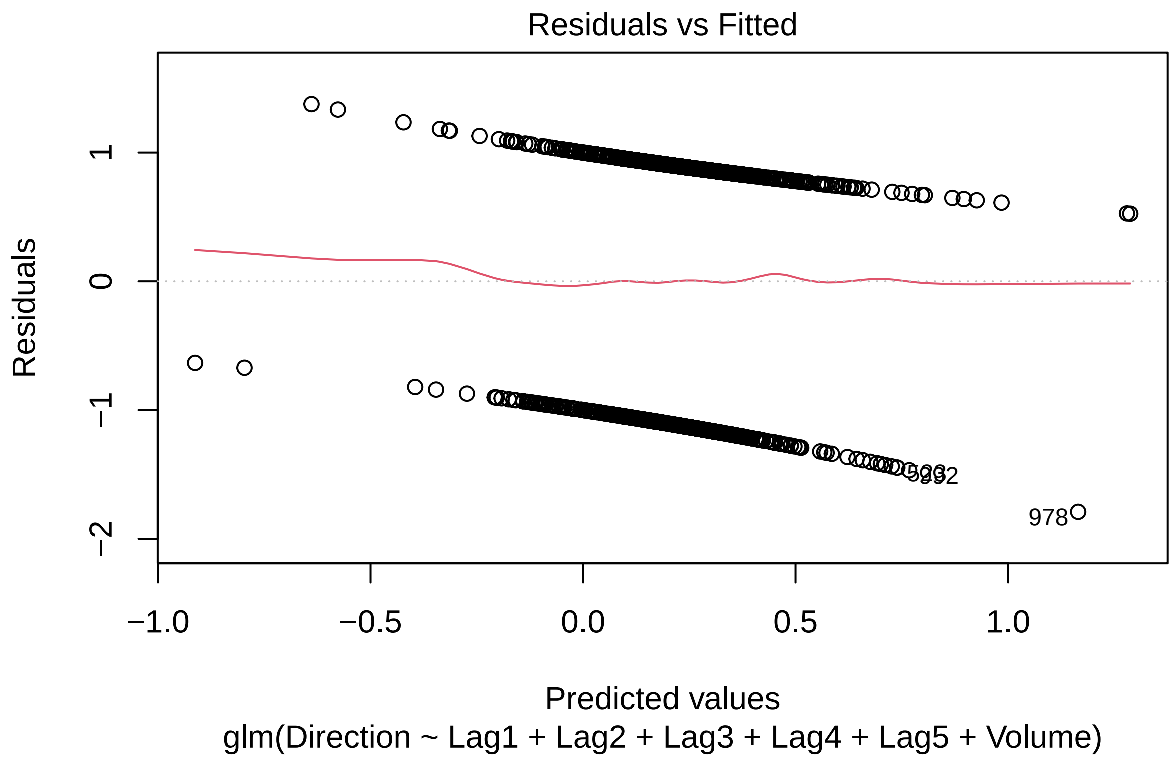


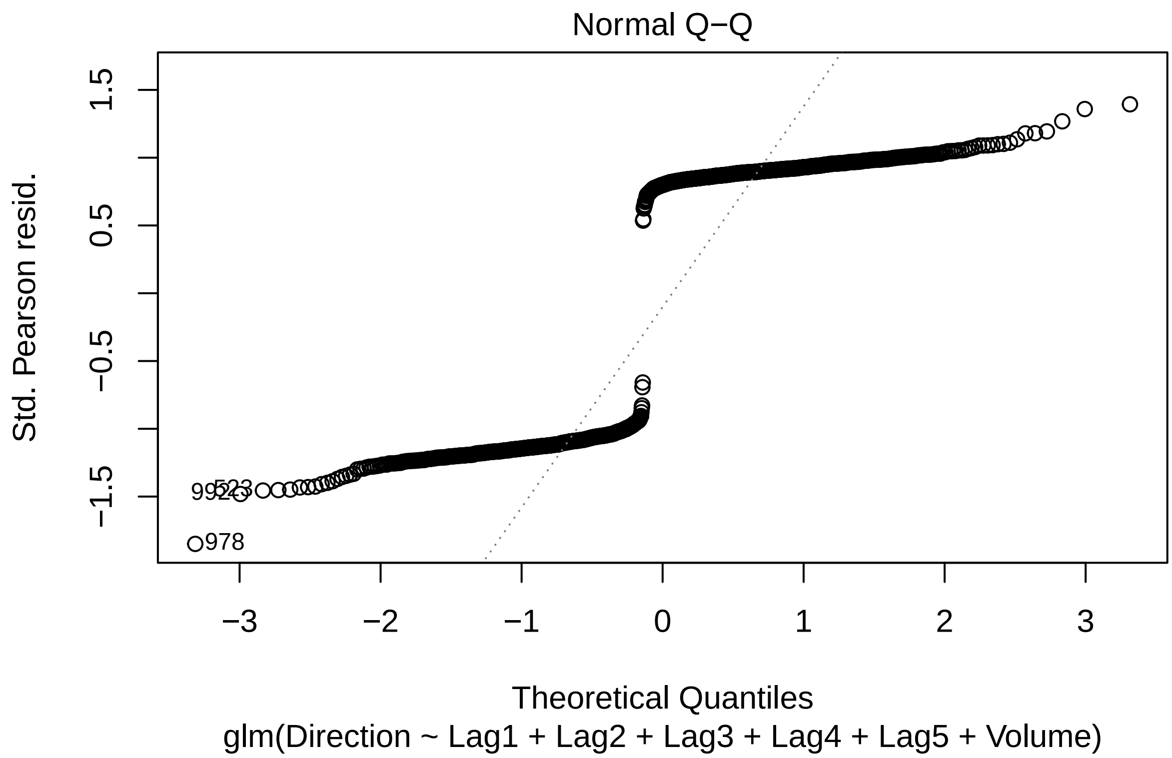
High degree of correlation between trading volume and year. Signifies increase in trading volume over the years, then dropping after 2008.

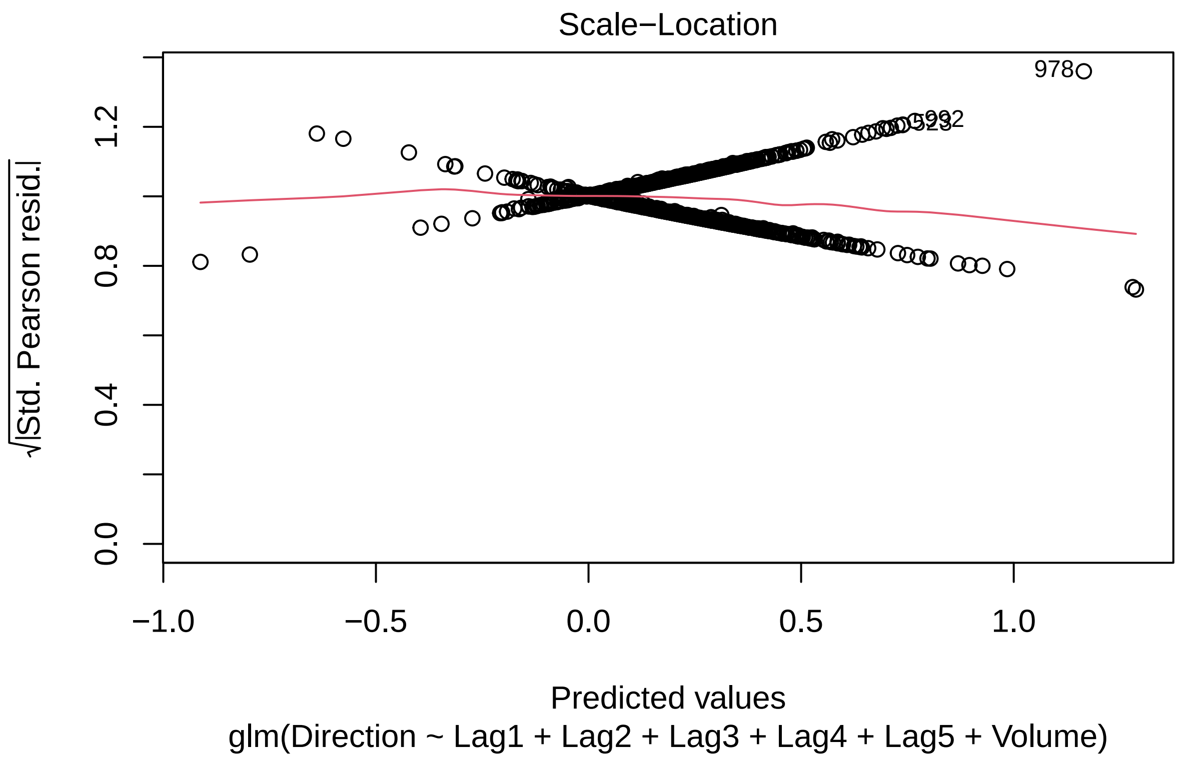
Part B

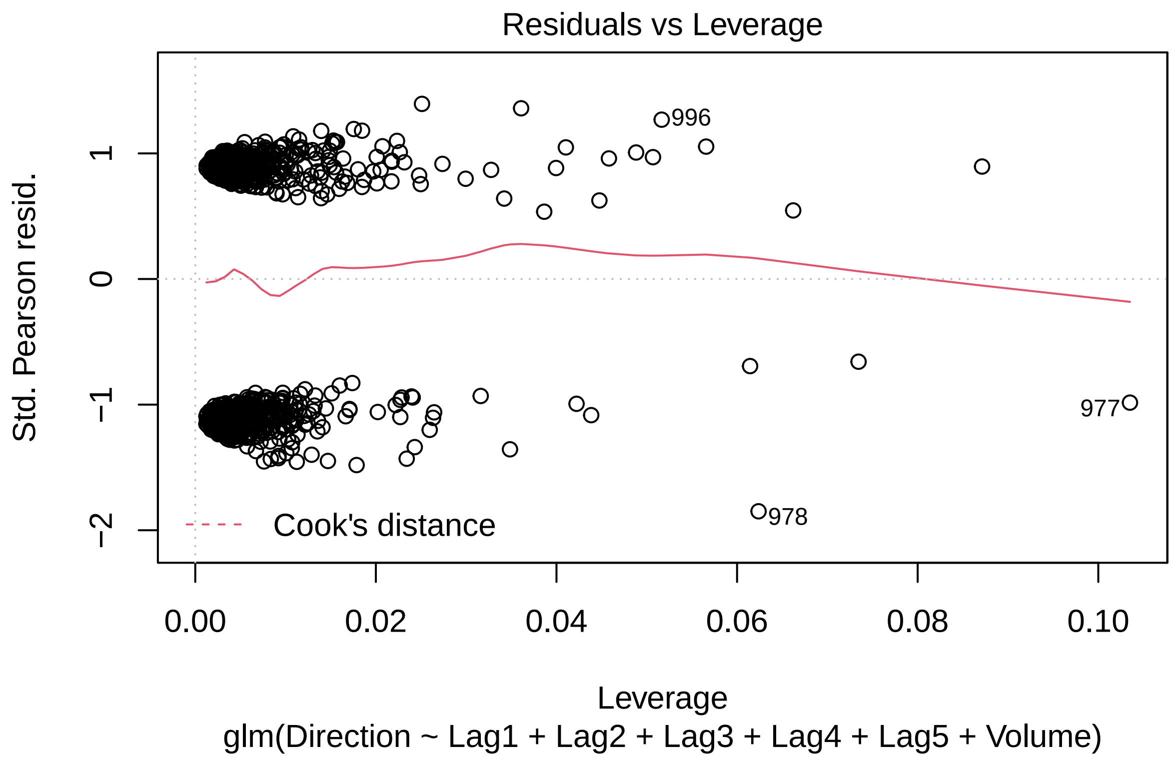
```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```




Normal Q−Q
Std. Pearson resid.
1.5
0.5
−0.5
−1.5
992
523
978
−3
−2
−1
0
1
2
3
Theoretical Quantiles
glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume)


Scale−Location
978
1292
523
√|Std. Pearson resid.|
0.0
0.4
0.8
1.2
−1.0
−0.5
0.0
0.5
1.0
Predicted values
glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume)




```
## (Intercept)        Lag1        Lag2        Lag3        Lag4        Lag5
##  0.26686414 -0.04126894  0.05844168 -0.01606114 -0.02779021 -0.01447206
##      Volume
## -0.02274153

##                Estimate Std. Error    z value    Pr(>|z|)
## (Intercept)  0.26686414 0.08592961  3.1056134 0.001898848
## Lag1        -0.04126894 0.02641026 -1.5626099 0.118144368
## Lag2         0.05844168 0.02686499  2.1753839 0.029601361
## Lag3        -0.01606114 0.02666299 -0.6023760 0.546923890
## Lag4        -0.02779021 0.02646332 -1.0501409 0.293653342
## Lag5        -0.01447206 0.02638478 -0.5485006 0.583348244
## Volume      -0.02274153 0.03689812 -0.6163330 0.537674762

## (Intercept)        Lag1        Lag2        Lag3        Lag4        Lag5
## 0.001898848 0.118144368 0.029601361 0.546923890 0.293653342 0.583348244
##      Volume
## 0.537674762
```

Lag2 has the lowest P value of 0.0296, and has a positive co-efficient but is the only

p-value outside the 95% confidence interval. Therefore the null hypothesis of returns being impacted by Lag2 is rejected.

Part C

```
##         1         2         3         4         5         6         7         8
## 0.6086249 0.6010314 0.5875699 0.4816416 0.6169013 0.5684190 0.5786097 0.5151972
##         9        10
## 0.5715200 0.5554287
```

```
##      Up
## Down  0
## Up    1
```

```
##         Direction
## glm.pred Down  Up
##     Down   54  48
##     Up    430 557
```

```
## [1] 0.5610652
```

Fraction of correct predictions: 0.561062 i.e (611/1089) 92% accuracy for up predictions (557/605) 11% accuracy for down predictions (54/484) Training error rate is 43.9% (100-56.1) False positive rate is 88.8% (430/484)

Part D

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = Weekly,
##     subset = d.train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.536  -1.264   1.021   1.091   1.368
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20326    0.06428   3.162  0.00157 **
## Lag2         0.05810    0.02870   2.024  0.04298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1354.5
##
## Number of Fisher Scoring iterations: 4
```

```
##
## glm.pred2 Down Up
##      Down    9  5
##      Up     34 56
```

```
## [1] 0.625
```

```
## [1] 0.5865385
```

Overall fraction of correct predictions is 62.5%

Part G

```
##
## knn.pred Down Up
##     Down   21 30
##     Up     22 31
```

```
## [1] 0.5
```

The prediction accuracy is 50%

Part H

Logistic regression is better at predicting stock direction data than knn is because the prediction accuracy for Logistic Regression is 62.5% while that for knn is 50%

Part I

Logistic regression with Lag1 to Lag4, from 1990 to 2000

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4, family = binomial,
##     data = Weekly, subset = d.train2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6051  -1.2735   0.9581   1.0544   1.4247
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.150e-01  8.931e-02   3.527 0.000421 ***
## Lag1        -1.032e-01  4.384e-02  -2.355 0.018524 *
## Lag2         2.747e-02  4.376e-02   0.628 0.530202
## Lag3        -2.629e-05  4.354e-02  -0.001 0.999518
## Lag4        -3.090e-02  4.339e-02  -0.712 0.476344
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 776.11  on 567  degrees of freedom
## Residual deviance: 768.88  on 563  degrees of freedom
## AIC: 778.88
##
## Number of Fisher Scoring iterations: 4
```

```
##
## glm.pred3 Down  Up
##      Down   22  25
##      Up    218 256

## [1] 0.5335893
```

Prediction accuracy is 53.36%

Logistic regression with Lag1 and Volume, from 1990 to 2000

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Volume, family = binomial, data = Weekly,
##     subset = d.train2)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.515  -1.287   0.956   1.051   1.434
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.44495    0.16188   2.749  0.00599 **
## Lag1        -0.10718    0.04318  -2.482  0.01305 *
## Volume      -0.29240    0.30723  -0.952  0.34124
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 776.11  on 567  degrees of freedom
## Residual deviance: 768.85  on 565  degrees of freedom
## AIC: 774.85
##
## Number of Fisher Scoring iterations: 4

##
## glm.pred4 Down  Up
##      Down  171 211
##      Up     69  70

## [1] 0.462572
```

Prediction accuracy is 46.257%

Variables with the best prediction accuracy is Lag2 and Volume, using Logistic regression

knn = 5

```
##
## knn.pred2 Down  Up
##      Down   82  98
##      Up    158 183
```

```
## [1] 0.5086372
```

Prediction accuracy 50.8637%

knn = 7

```
##
## knn.pred3 Down  Up
##      Down   73  91
##      Up    167 190
```

```
## [1] 0.5047985
```

Prediction accuracy 50.4799%

knn = 10

```
##
## knn.pred4 Down  Up
##      Down   76  93
##      Up    164 188
```

```
## [1] 0.5067179
```

Prediction accuracy 50.6718%

Chapter 6 Problem 9

Part A

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-2
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
## [1] 388  18
```

Dimensions of the training set are: 388 rows, 18 columns

Part B

```
##
## Call:
## lm(formula = Apps ~ ., data = c.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24771 -0.10729 -0.00961  0.07873  1.91283
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.075438   0.040377   1.868 0.062501 .
## PrivateYes  -0.112000   0.051793  -2.162 0.031223 *
## Accept       1.039178   0.031750  32.730  < 2e-16 ***
## Enroll      -0.364014   0.061913  -5.879 9.20e-09 ***
## Top10perc    0.175092   0.038197   4.584 6.25e-06 ***
## Top25perc   -0.046150   0.034866  -1.324 0.186440
## F.Undergrad  0.195182   0.057994   3.366 0.000844 ***
## P.Undergrad  0.020588   0.015176   1.357 0.175748
## Outstate    -0.089697   0.029222  -3.069 0.002303 **
## Room.Board   0.042935   0.019643   2.186 0.029461 *
## Books        0.005947   0.018955   0.314 0.753913
## Personal     0.003716   0.014751   0.252 0.801260
## PhD         -0.041988   0.031698  -1.325 0.186104
## Terminal    -0.013124   0.031291  -0.419 0.675142
## S.F.Ratio    0.047025   0.021869   2.150 0.032182 *
## perc.alumni  0.012615   0.018825   0.670 0.503206
## Expend       0.131604   0.029728   4.427 1.26e-05 ***
## Grad.Rate    0.028126   0.018893   1.489 0.137406
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2793 on 370 degrees of freedom
## Multiple R-squared:  0.9432, Adjusted R-squared:  0.9406
## F-statistic: 361.3 on 17 and 370 DF,  p-value: < 2.2e-16
```

```
## [1] 0.07301206
```

Test MSE is 0.07301206

Part C

```
## [1] 0.01
```

```
## [1] 0.07042784
```

Lambda value is 0.01 Test MSE is 0.0704278

Part D

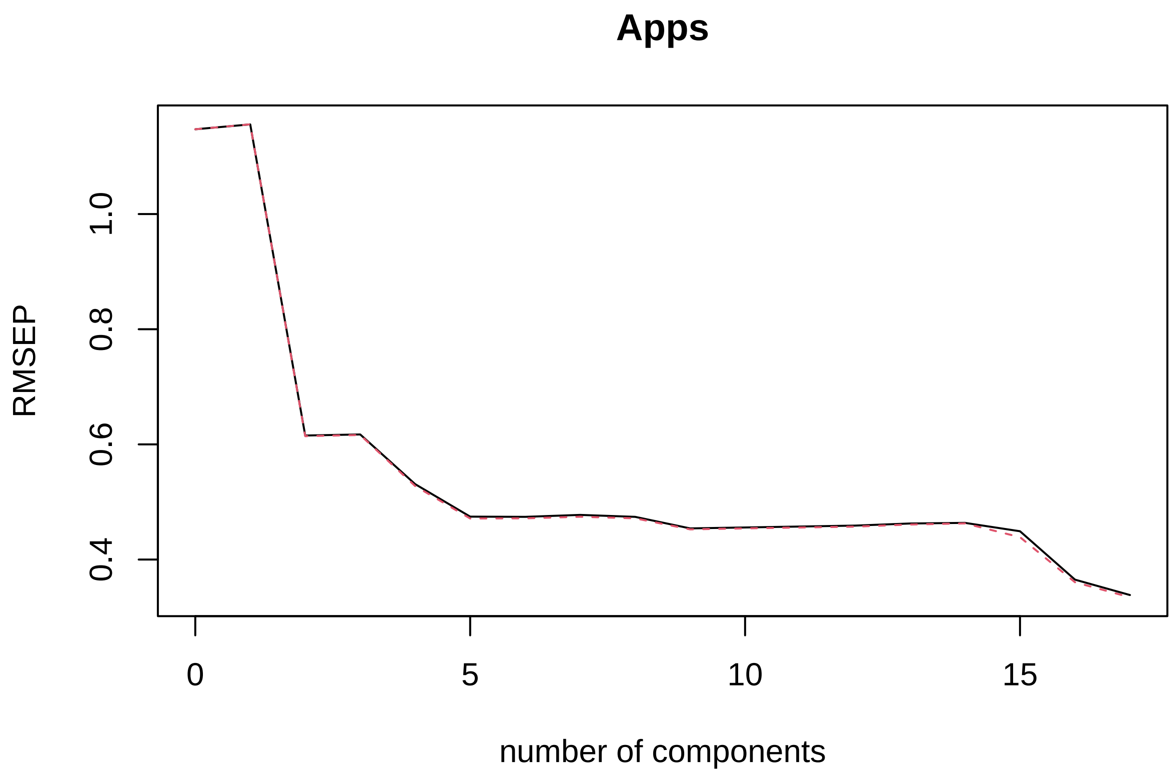
```
## [1] 0.0231013
```

```
## [1] 0.07616939

## 19 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) -1.494915e-02
## PrivateNo    5.479006e-02
## PrivateYes  -4.293745e-14
## Accept       8.799119e-01
## Enroll       .
## Top10perc    1.110172e-01
## Top25perc    .
## F.Undergrad  .
## P.Undergrad  .
## Outstate     .
## Room.Board   .
## Books        .
## Personal     .
## PhD          .
## Terminal     .
## S.F.Ratio    .
## perc.alumni  .
## Expend       5.967475e-02
## Grad.Rate    .
```

Lambda value is 0.0231013 Test MSE is 0.07616939

Part E

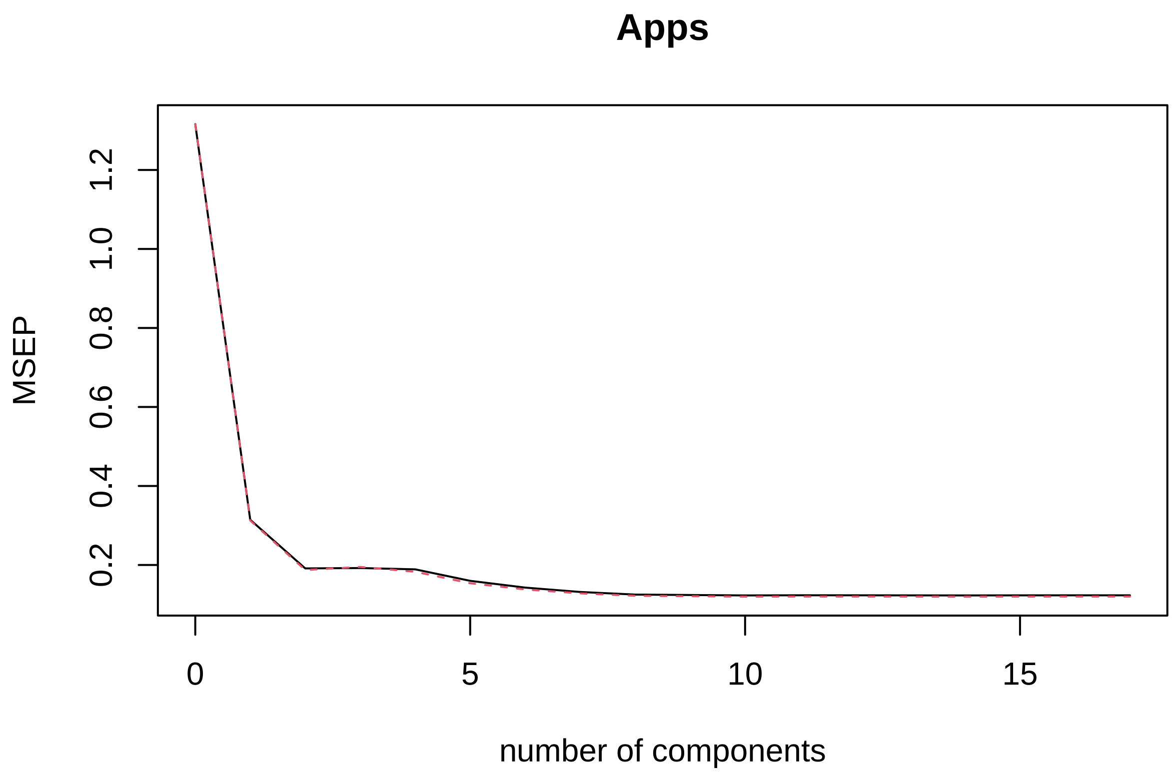
```
## Warning in plot.window(...): "valtype" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "valtype" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "valtype" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "valtype" is not a
## graphical parameter

## Warning in box(...): "valtype" is not a graphical parameter

## Warning in title(...): "valtype" is not a graphical parameter
```




```
## [1] 0.06782108
```

MSE is 0.06782283 M=16 gives us the lowest cross validation error

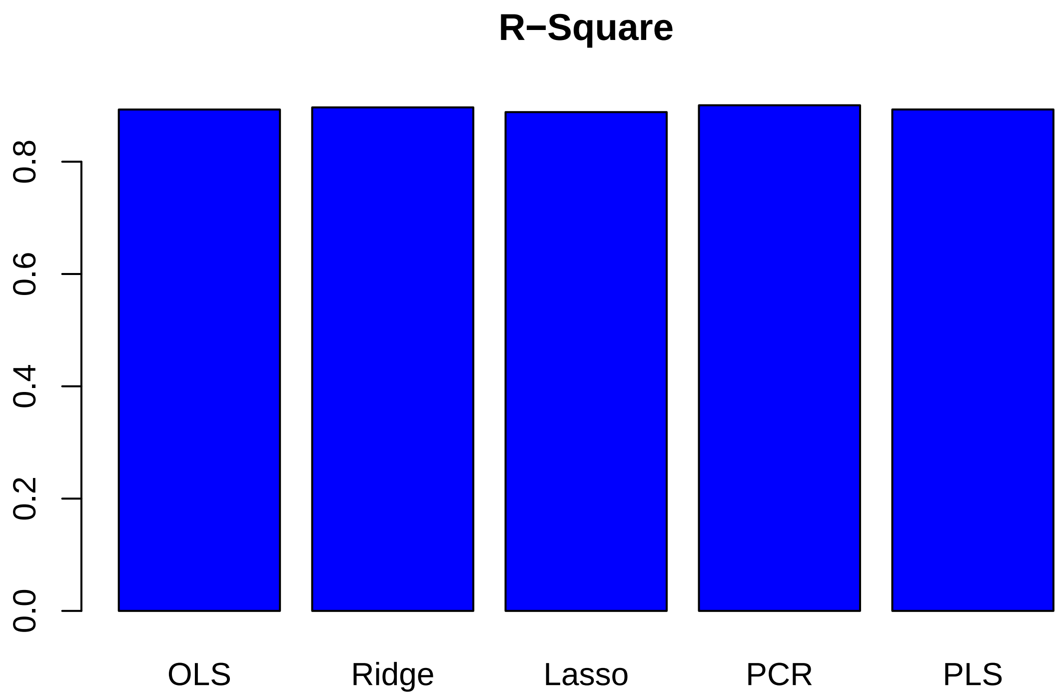
Part F




```
## [1] 0.07294478
```

MSE is 0.07172876 M=10 gives us the lowest cross validation error

Part G



Ridge MSE = 0.0704278 Lasso MSE = 0.07616939 Least Squares MSE = 0.07301206 PCR MSE = 0.06782283 PLS MSE = 0.07172876 Therefore, the best fit model is PCR, however the R squared are marginally different.

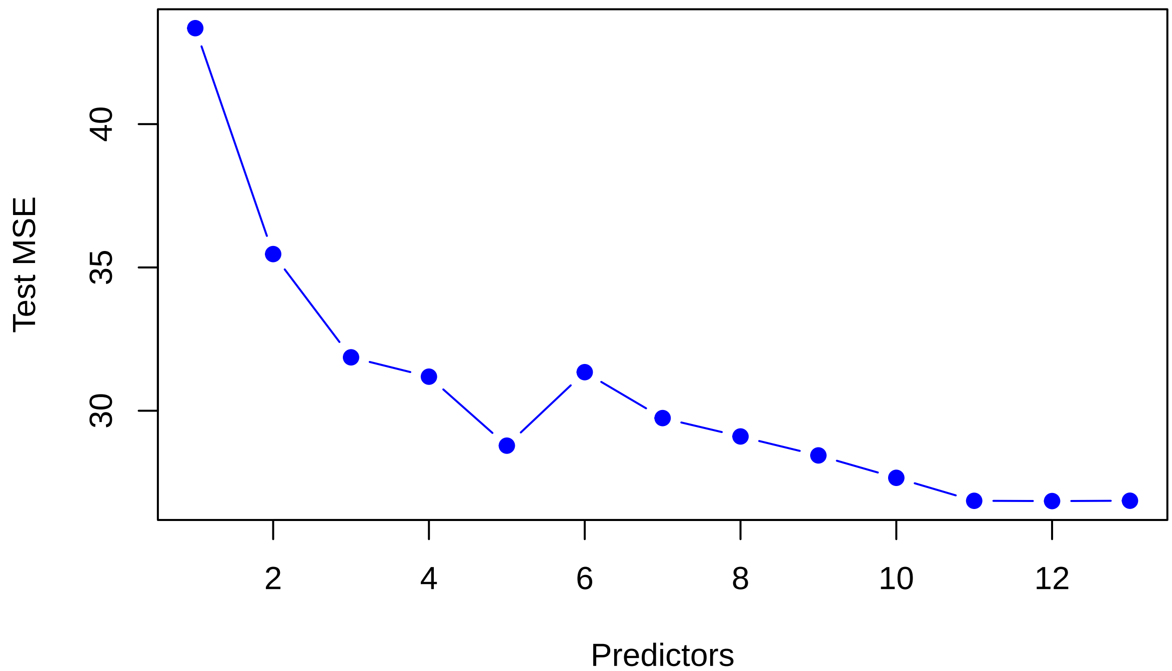
Chapter 6 Problem 11

Part A

```
## [1] 253  14
```

Dimensions of the training set is: 253 rows 14 columns

Best Subset Selection



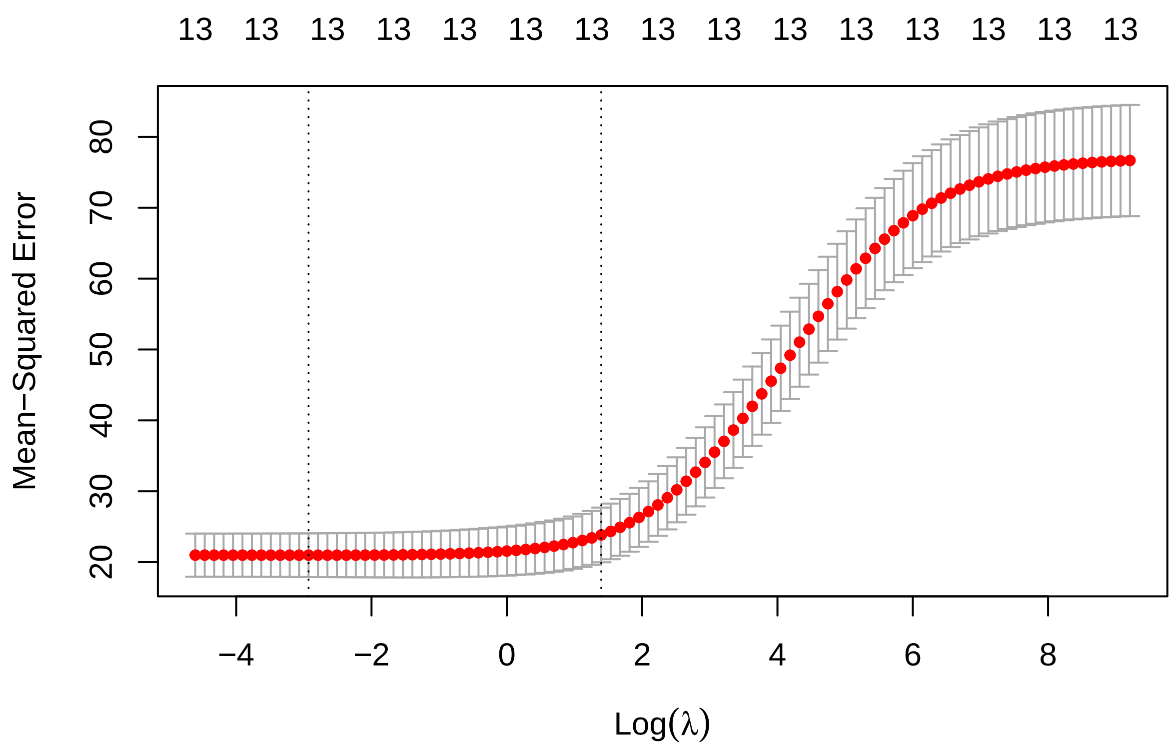

```
## [1] 12
```

```
##   (Intercept)          crim            zn         indus          chas
##  34.831120307  -0.091918101   0.030237800   0.033374023   2.295987544
##           nox            rm           dis           rad           tax
## -13.811099542   4.079128964  -1.244699757   0.382084606  -0.019216474
##       ptratio         black         lstat
##  -0.985922901   0.006773852  -0.492341180
```

```
## [1] 26.84946
```

MSE is 26.84946

Lasso



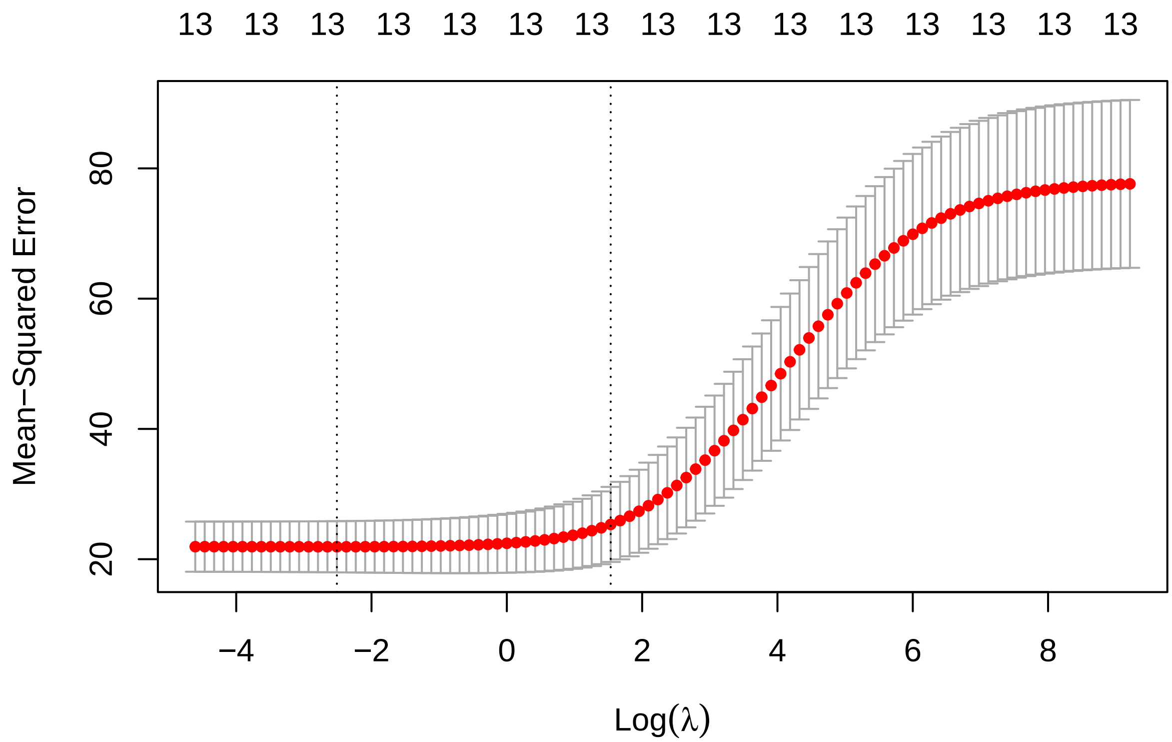

```
## [1] 0.05336699
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  33.549938944
## (Intercept)   .
## crim         -0.089333575
## zn            0.028630555
## indus         0.022489614
## chas          2.372039925
## nox         -13.053954757
## rm            4.138827718
## age          -0.004384557
## dis          -1.236522044
## rad           0.347615984
## tax          -0.017518196
## ptratio      -0.965923891
## black         0.006772119
## lstat        -0.481050642
```

```
## [1] 26.79522
```

Lambda is 0.01 MSE is 26.79522

Ridge




```
## [1] 0.08111308
```

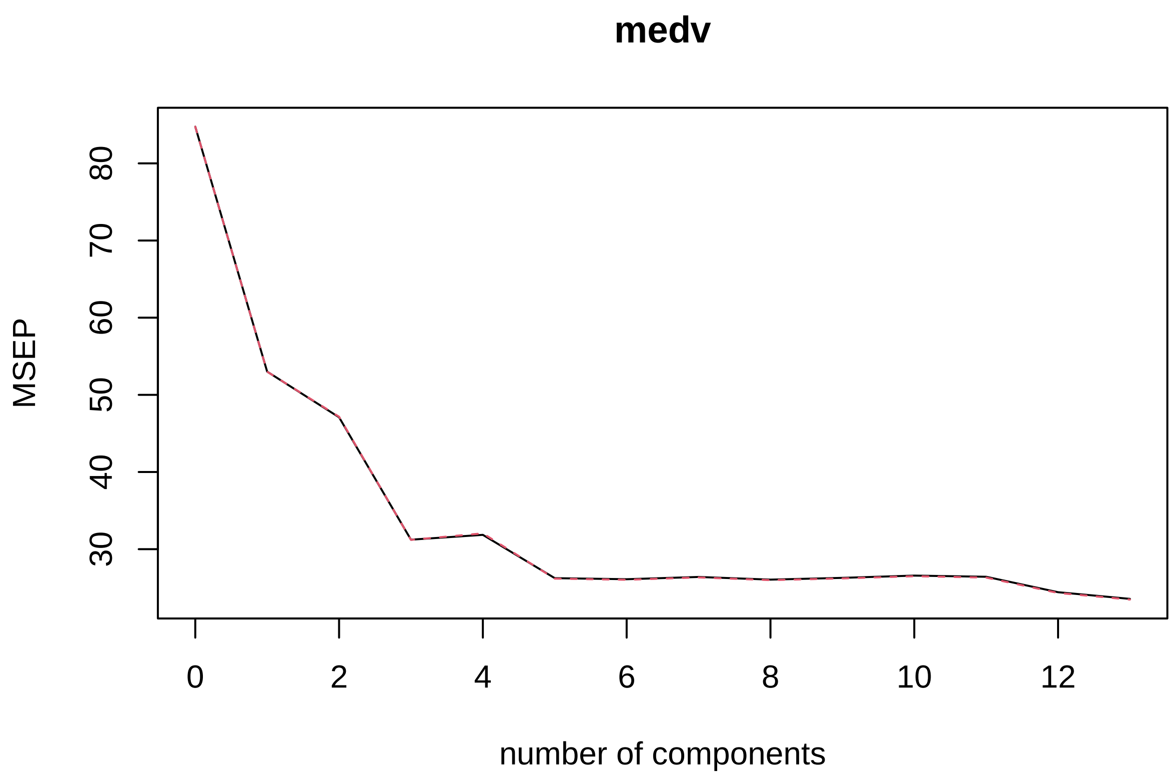
```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  33.024094010
## (Intercept)   .
## crim         -0.088159911
## zn            0.027992223
## indus         0.018467094
## chas          2.400885843
## nox         -12.807002261
## rm            4.155104180
## age          -0.004805140
## dis          -1.221950655
## rad           0.333681186
## tax          -0.016820543
## ptratio      -0.959305356
## black         0.006760755
## lstat        -0.478261284
```

```
## [1] 26.77717
```

Lambda = 0.01 Ridge MSE is 26.77717

PCR

```
## Data:    X dimension: 506 13
##  Y dimension: 506 1
## Fit method: svdpc
## Number of components considered: 13
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           9.206    7.281    6.862    5.588    5.644    5.123    5.109
## adjCV        9.206    7.280    6.868    5.587    5.661    5.117    5.102
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       5.138    5.104    5.127     5.155      5.14     4.942     4.854
## adjCV    5.132    5.098    5.120     5.147      5.13     4.932     4.845
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X       47.13    58.15    67.71    74.31    80.73    85.79    89.91    92.95
## medv    37.42    45.59    63.59    64.78    69.70    70.05    70.05    70.56
##       9 comps  10 comps  11 comps  12 comps  13 comps
## X       95.08     96.78     98.21     99.51    100.00
## medv    70.57     70.89     71.30     73.21     74.06
```




```
## Data:    X dimension: 253 13
```

```
##  Y dimension: 253 1
## Fit method: svdpc
## Number of components considered: 5
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps
## X       48.02    58.80    67.99    75.11    80.85
## medv    40.83    43.75    64.55    72.78    72.81

## , , 1 comps
##
##                medv
## crim    -0.58901583
## zn       0.55290872
## indus   -0.75658758
## chas     0.04169713
## nox     -0.77186413
## rm       0.42824335
## age     -0.69998703
## dis      0.72258349
## rad     -0.71757547
## tax     -0.75690381
## ptratio -0.48362619
## black    0.48103143
## lstat   -0.68991293
##
## , , 2 comps
##
##               medv
## crim    -1.0327137
## zn       0.1183425
## indus   -0.5461255
## chas     0.5908909
## nox     -0.5331747
## rm       0.4320955
## age     -0.3325371
## dis      0.3178432
## rad     -1.0960500
## tax     -1.0634332
## ptratio -0.7103027
## black    0.9657038
## lstat   -0.6915826
##
## , , 3 comps
##
##                medv
## crim    -0.41021087
## zn       1.06227200
## indus   -0.44361778
## chas     2.25645994
## nox      0.04055328
## rm       2.48038051
## age     -0.31224947
## dis     -0.15731819
## rad     -0.18301087
```

```
## tax     -0.34805561
## ptratio -2.08747479
## black    0.25720650
## lstat   -1.63011220
##
## , , 4 comps
##
##                medv
## crim    -0.94766845
## zn       0.14082804
## indus   -0.37620155
## chas     0.97649527
## nox      0.02713317
## rm       3.94426169
## age     -0.14133317
## dis     -0.64803172
## rad      0.03854496
## tax     -0.18770194
## ptratio -1.34190432
## black    0.43537465
## lstat   -2.62947540
##
## , , 5 comps
##
##                 medv
## crim    -0.936993259
## zn       0.161504585
## indus   -0.364860941
## chas     0.864967231
## nox      0.063304433
## rm       3.952029786
## age     -0.105517898
## dis     -0.675079851
## rad     -0.005581025
## tax     -0.214616117
## ptratio -1.471135730
## black    0.405716567
## lstat   -2.596174401
```

```
## [1] 30.16412
```

Test MSE is 30.16412

Part B

The Validation MSEs are as follows: Subset: 26.79522 Lasso: 26.89298 Ridge: 26.77717 PCR: 30.16412 The Ridge model appears to best predict the median housing data, since the MSE for Ridge is the lowest. The MSE for lasso is a close second.

Part C

The Ridge regression model contains all the variables in the data set.Therefore all variables contribute to prediction of the target variable.

Word Problem 1 - Beauty Data

Part A

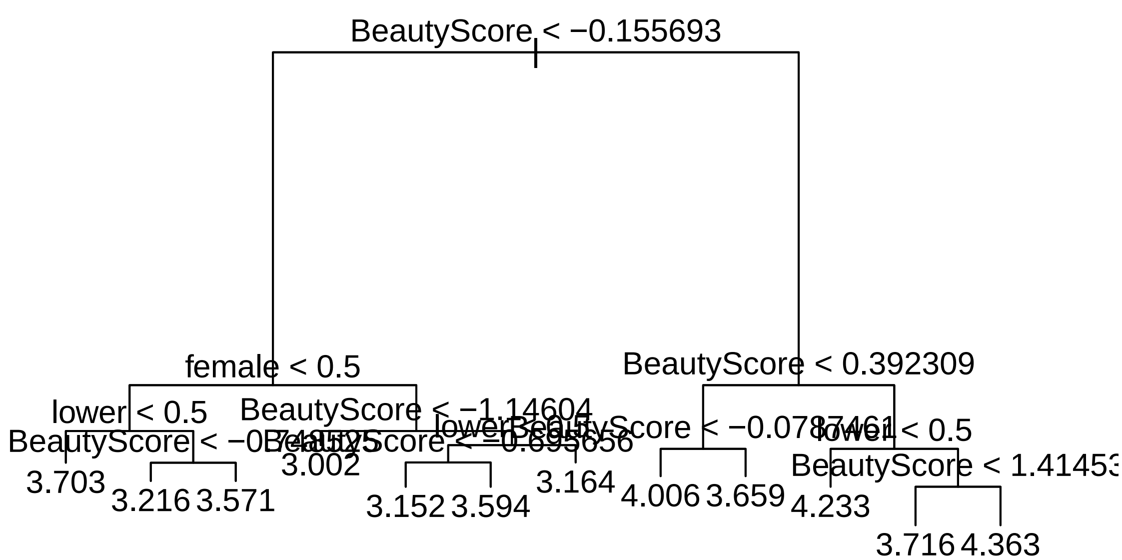
With Beauty

```
## [1] 231   6
```

Dimensions of the training set is: 231 rows 6 columns

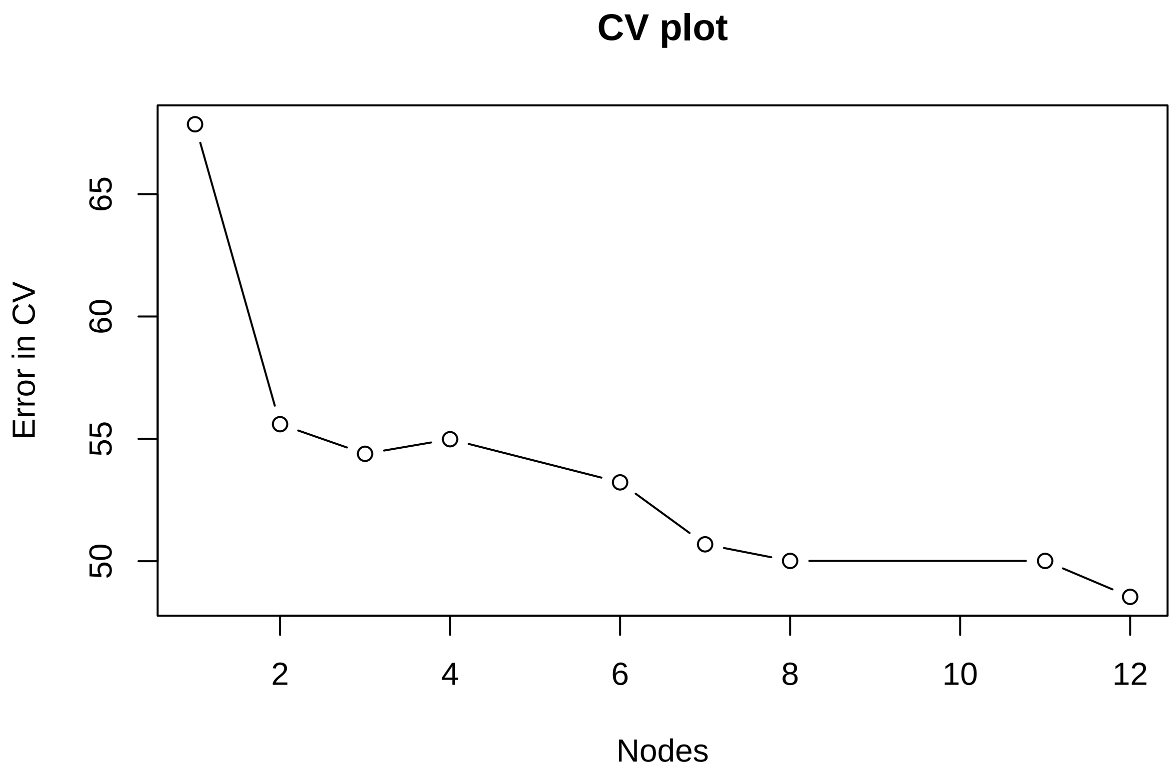
Decision Trees

```
##
## Regression tree:
## tree(formula = CourseEvals ~ ., data = train)
## Variables actually used in tree construction:
## [1] "BeautyScore" "female"      "lower"
## Number of terminal nodes:  12
## Residual mean deviance:  0.1624 = 35.57 / 219
## Distribution of residuals:
##       Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.103000 -0.290400 -0.007418  0.000000  0.228200  1.154000
```



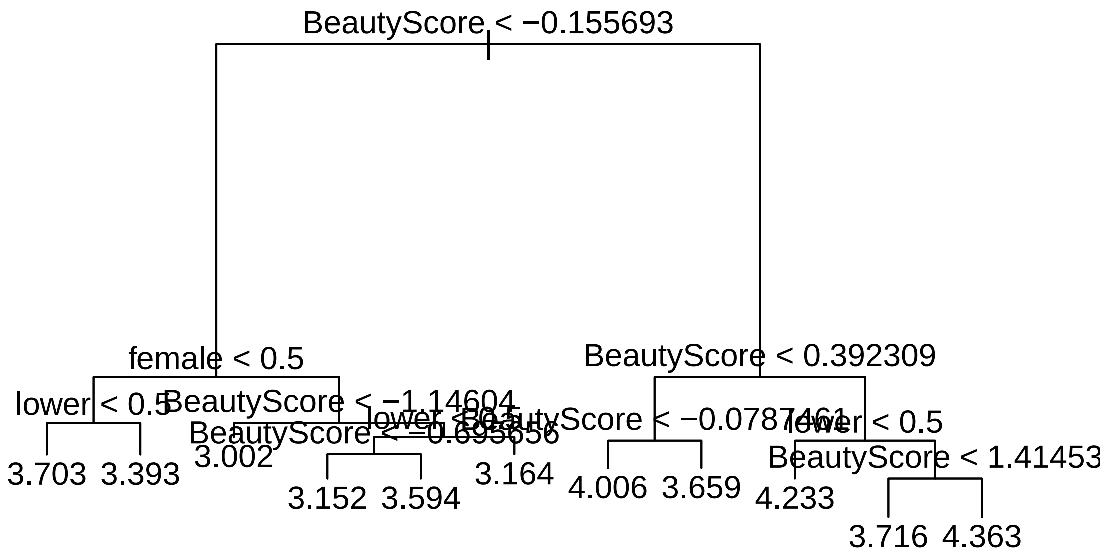

```
## [1] 0.253748
```

Test MSE 0.2300245



Cross Validation

```
## $size
## [1] 12 11  8  7  6  4  3  2  1
##
## $dev
## [1] 48.54354 50.01246 50.01246 50.69106 53.22257 54.98561 54.38876 55.60027
## [9] 67.85522
##
## $k
## [1]       -Inf  0.8813947  0.8980289  1.3195882  1.5311937  1.8445564  2.2153412
## [8]  3.0759961 16.0802980
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```



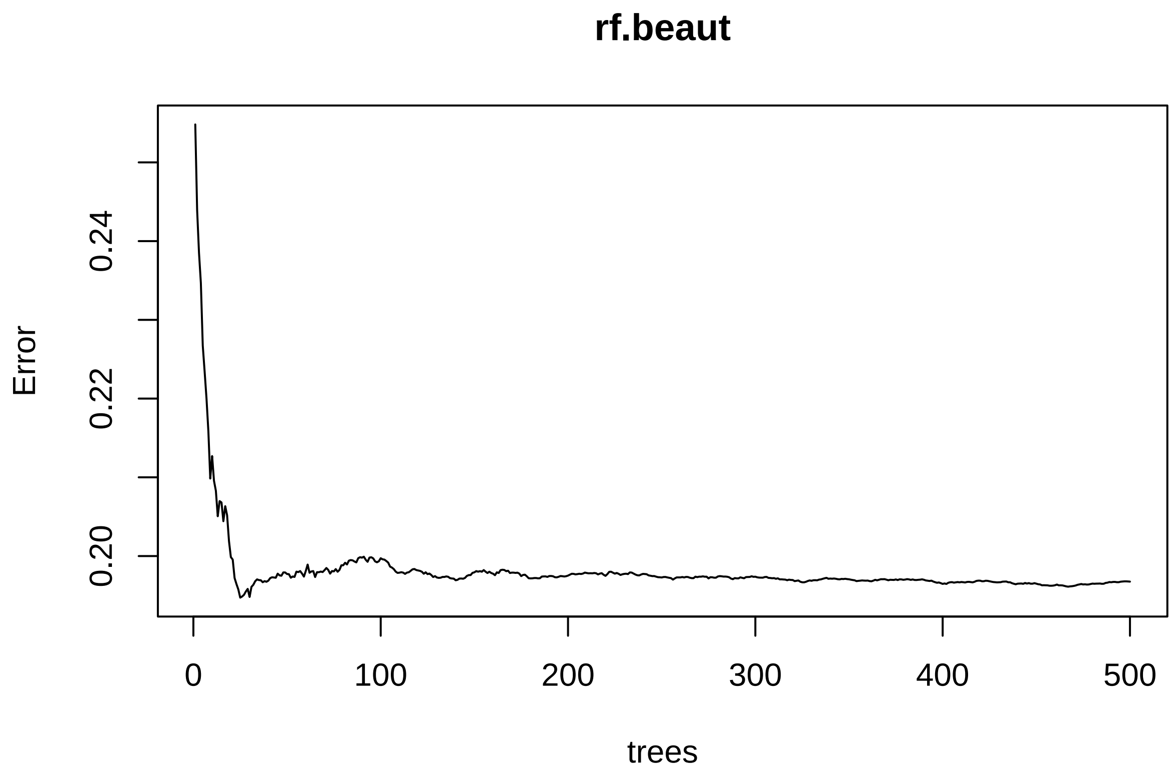
Pruning

```
## [1] 0.2551065
```

```
##   CourseEvals BeautyScore female lower nonenglish tenuretrack
## 1    3.235245   0.2015666      1     0          0           1
## 2    3.226328  -0.8260813      0     0          0           1
## 3    3.647712  -0.6603327      0     0          0           1
## 4    3.372062  -0.7663125      1     0          0           1
## 5    4.292705   1.4214450      1     0          0           1
## 6    4.239140   0.5002196      0     0          0           1
```

RandomForest

```
## [1] 0.2340579
```

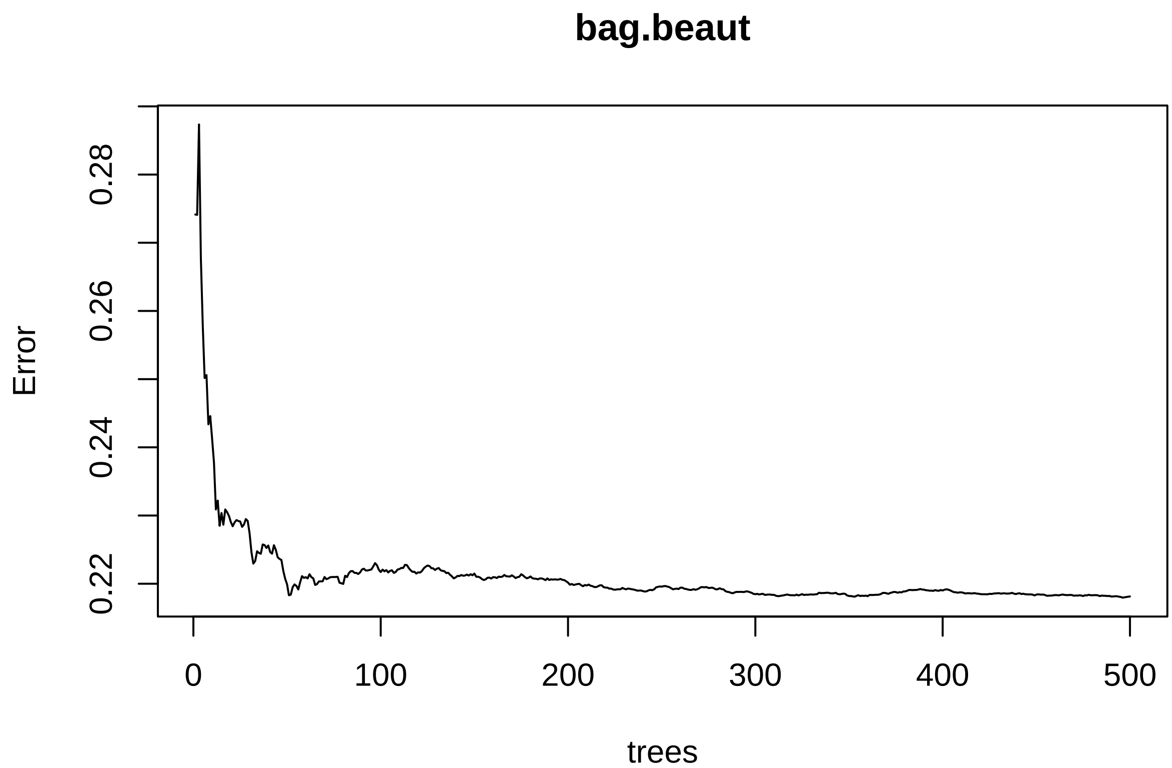



```
##              %IncMSE IncNodePurity
## BeautyScore 56.646852     31.163681
## female      22.797578      4.657400
## lower       36.754581      4.984673
## nonenglish   9.868252      1.586019
## tenuretrack  5.294083      1.849735
```

Random Forest MSE is 0.2276729

Bagging

```
## [1] 0.2548296
```

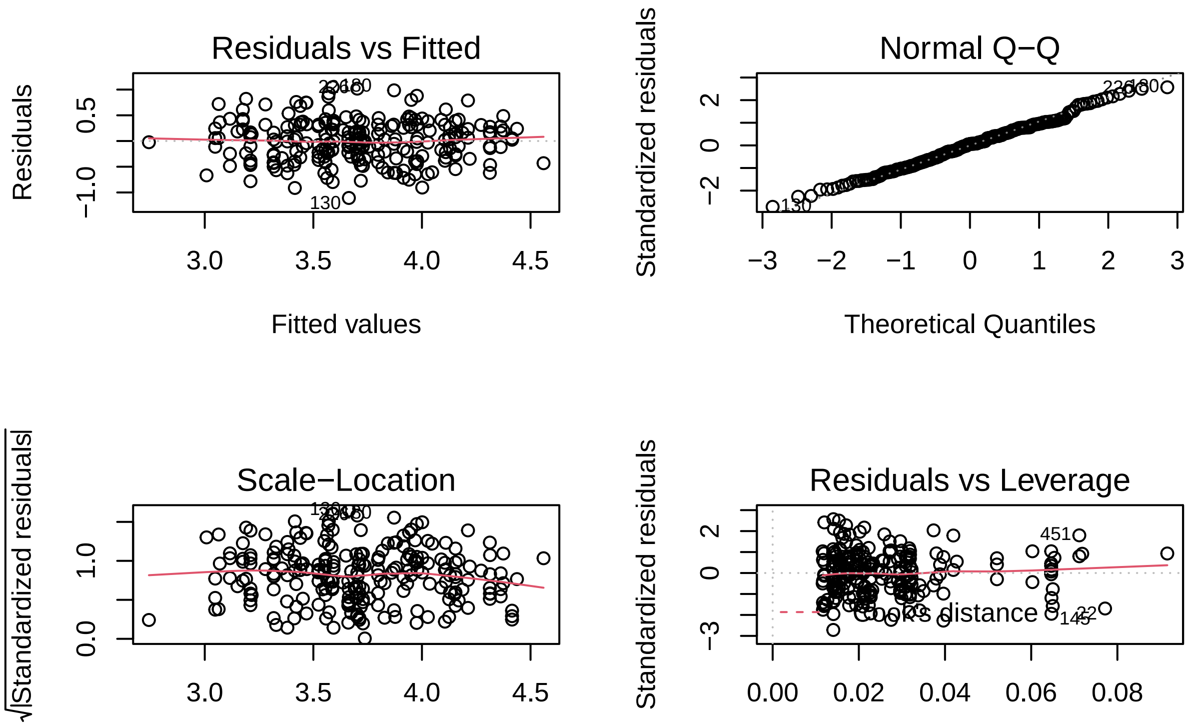


Bagging MSE is 0.2452319

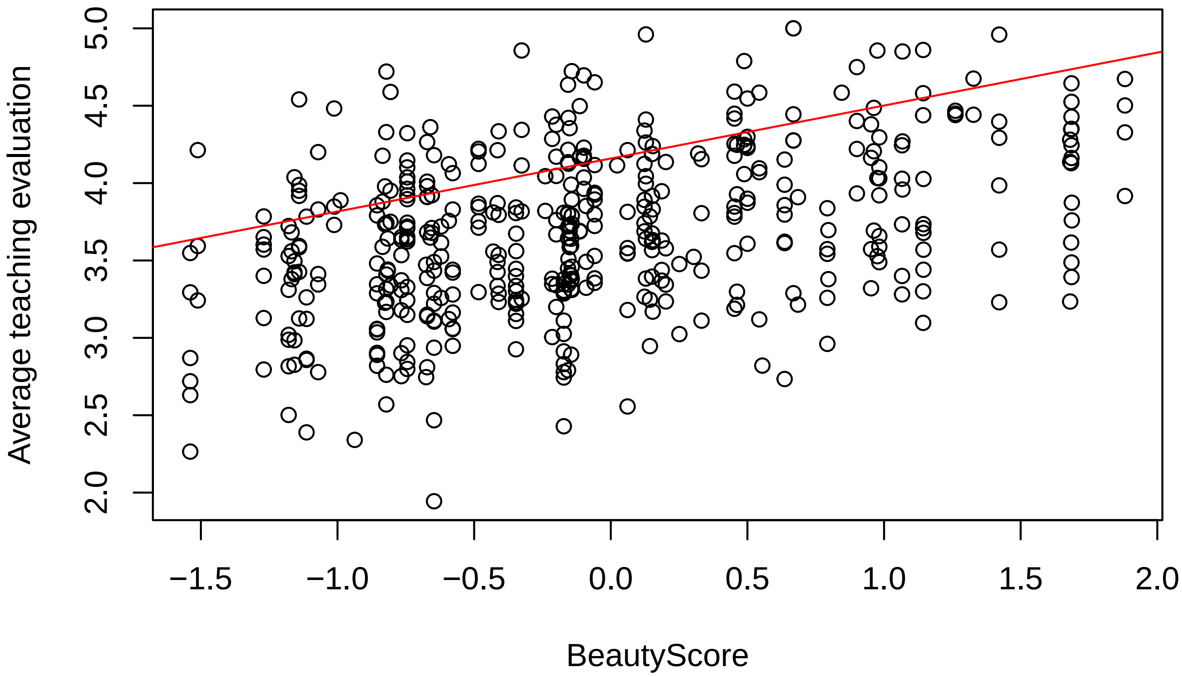
Linear regression

```
##
## Call:
## lm(formula = CourseEvals ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.10707 -0.29304  0.02471  0.28495  1.04659
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.15882    0.07073  58.802  < 2e-16 ***
## BeautyScore  0.34222    0.03449   9.922  < 2e-16 ***
## female      -0.34062    0.05535  -6.154 3.43e-09 ***
## lower       -0.37317    0.05825  -6.406 8.64e-10 ***
## nonenglish  -0.31369    0.10871  -2.886  0.00429 **
## tenuretrack -0.17557    0.06820  -2.574  0.01068 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4104 on 225 degrees of freedom
## Multiple R-squared:  0.4349, Adjusted R-squared:  0.4223
## F-statistic: 34.63 on 5 and 225 DF,  p-value: < 2.2e-16
```

```
##                   2.5 %      97.5 %
## (Intercept)  4.0194514  4.29819045
## BeautyScore  0.2742492  0.41018769
## female      -0.4496897 -0.23154873
## lower       -0.4879603 -0.25837274
## nonenglish  -0.5279118 -0.09946688
## tenuretrack -0.3099649 -0.04118342
```



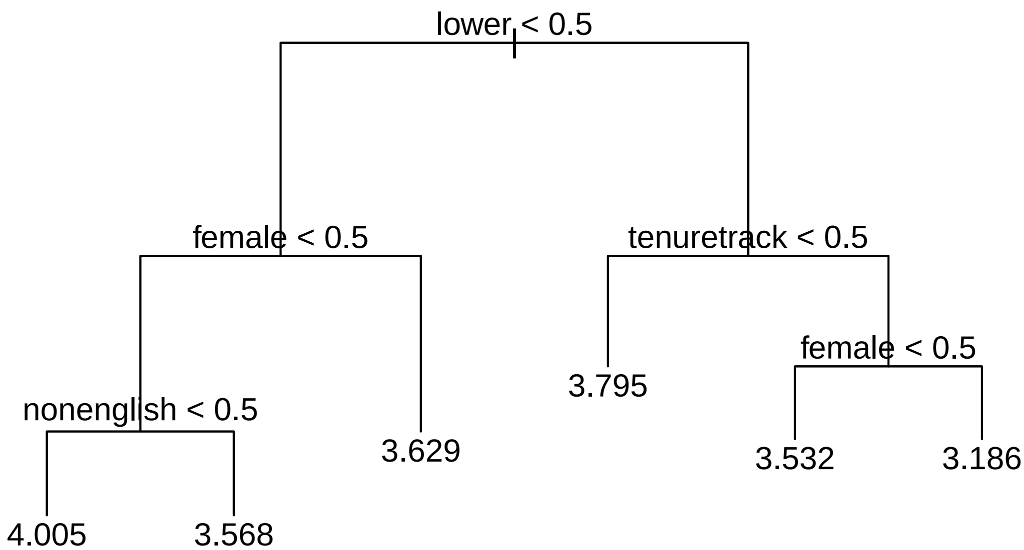

```
## Warning in abline(lm.fit1, col = "red"): only using the first two of 6
## regression coefficients
```



Without Beauty

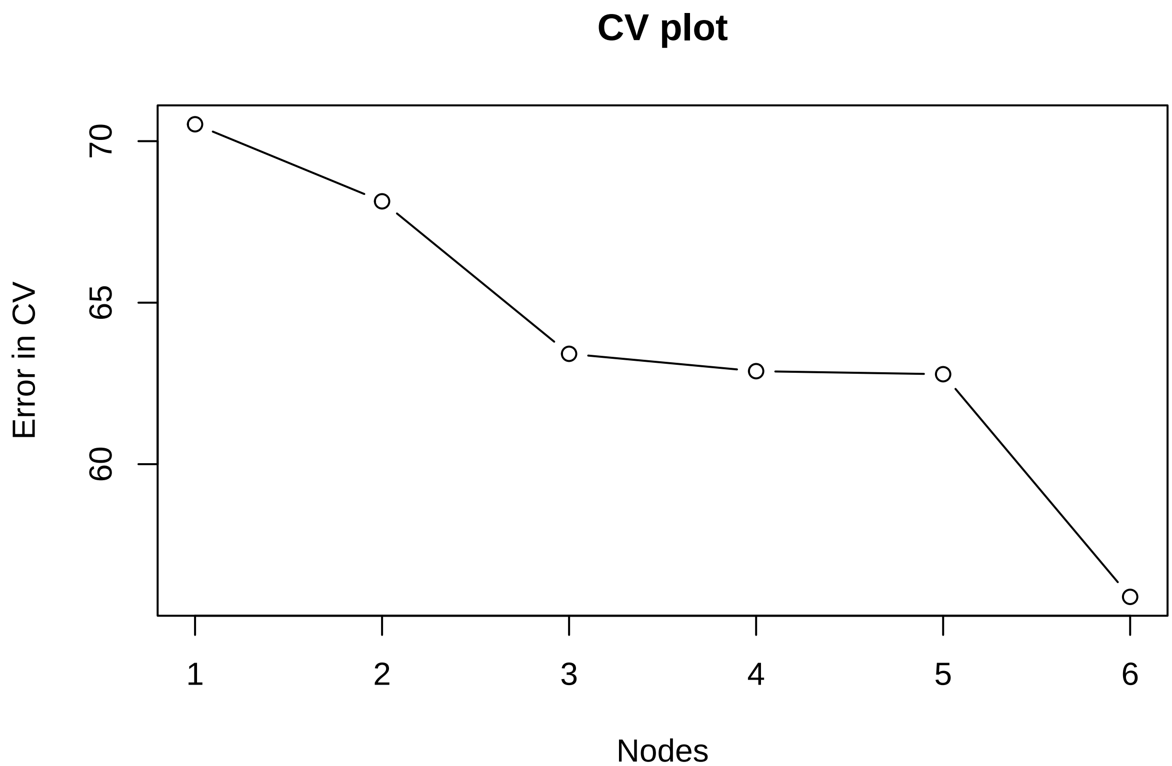
Decision Tree

```
##
## Regression tree:
## tree(formula = CourseEvals ~ . - BeautyScore, data = train)
## Number of terminal nodes:  6
## Residual mean deviance:  0.2343 = 52.71 / 225
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.288000 -0.325300 -0.009816  0.000000  0.315800  1.228000
```



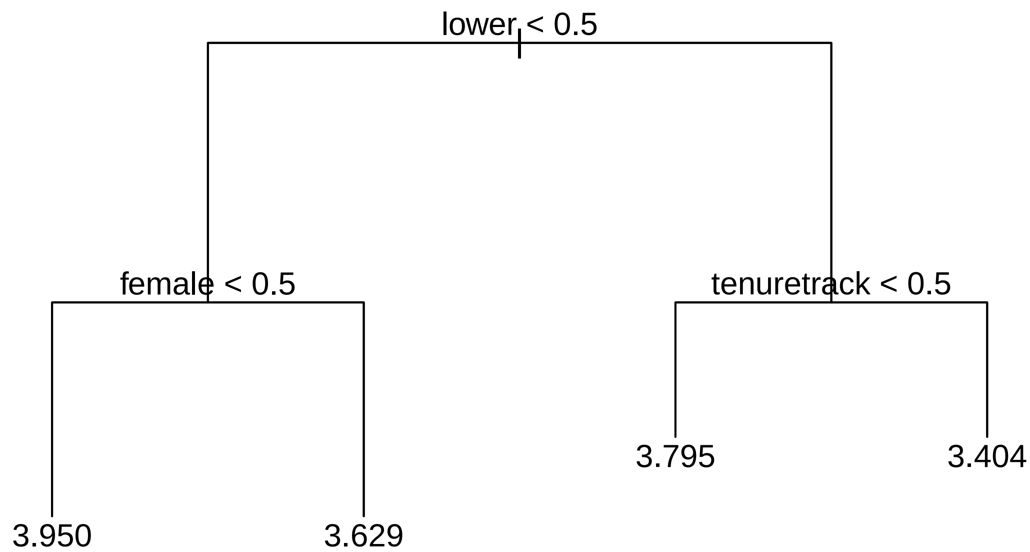

```
## [1] 0.2530681
```

Test MSE is 0.2942618



Cross Validation

```
## $size
## [1] 6 5 4 3 2 1
##
## $dev
## [1] 55.89262 62.78580 62.87929 63.41672 68.13775 70.52318
##
## $k
## [1]     -Inf 1.591635 1.834427 2.417605 3.843091 4.663997
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```



Pruning

```
## [1] 0.2451703
```

MSE is 0.2942618

Random Forests

```
## [1] 0.250945
```

Random Forest MSE is 0.2800344

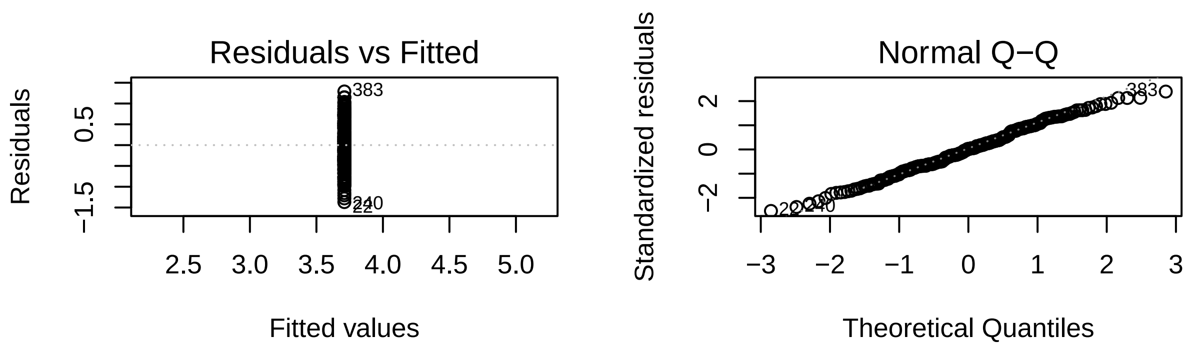
Linear regression

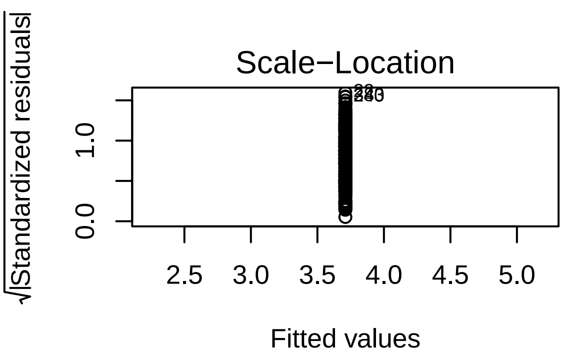
```
##
## Call:
## lm(formula = CourseEvals ~ -BeautyScore, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.36944 -0.36609  0.01243  0.41462  1.28991
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.71009    0.03553   104.4   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.54 on 230 degrees of freedom
```

```
##                2.5 %   97.5 %
## (Intercept) 3.640095 3.780094
```

```
## hat values (leverages) are all = 0.004329004
##  and there are no factor predictors; no plot no. 5
```





Removing the BeautyScore reduces the accuracy of the model by 34.68%. The test MSE without the BeautyScore is larger than that with the BeautyScore. The second most impactful variable is the gender.

Part B

Based of human nature, automatic biases and to control biases, have a control group and standardize the voice. Can't inherently quantify beauty, each person has different standards of beauty i.e subjective variable, therefore the other variables didn't have the intended effect. Because the eval cannot be explained without beauty, removing beauty out of the equation leads to a more complex model.

Word Problem 2 - Midcity Data

```
##       Home              Nbhd            Offers             SqFt
```

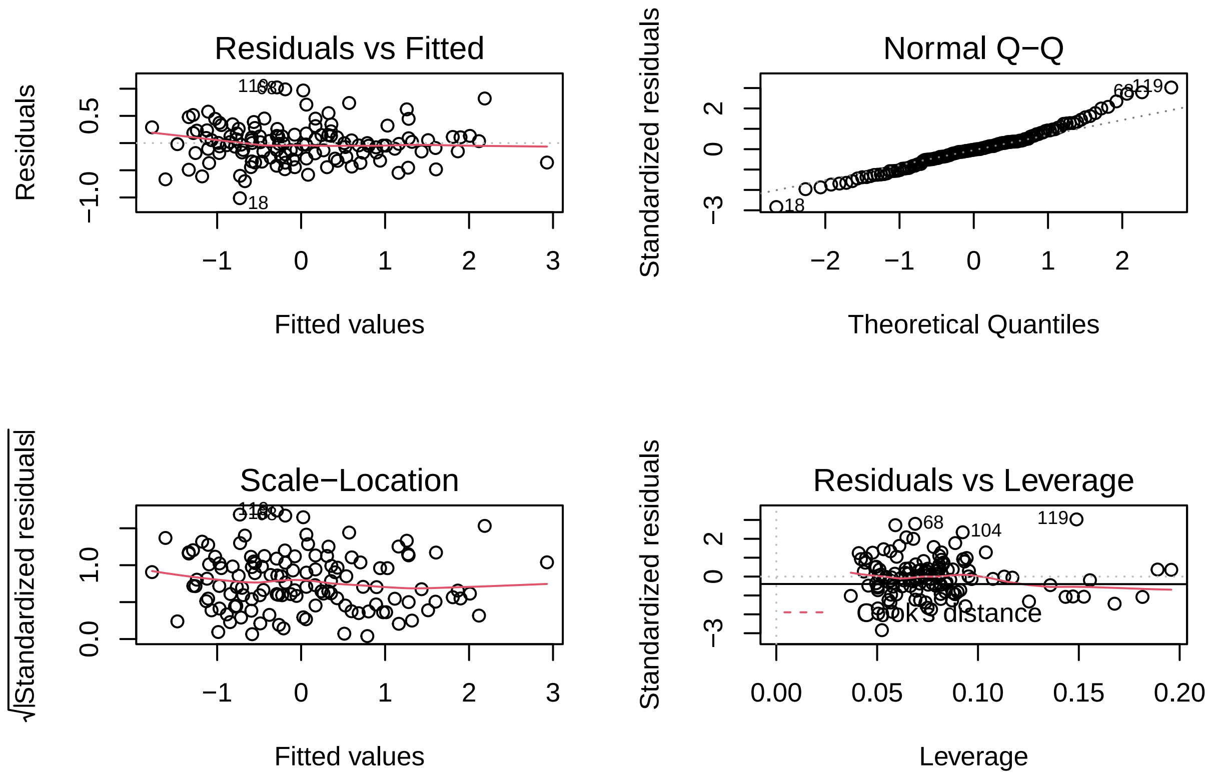
```
##  Min.   :  1.00   Min.   :1.000   Min.   :1.000   Min.   :1450
##  1st Qu.: 32.75   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1880
##  Median : 64.50   Median :2.000   Median :3.000   Median :2000
##  Mean   : 64.50   Mean   :1.961   Mean   :2.578   Mean   :2001
##  3rd Qu.: 96.25   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:2140
##  Max.   :128.00   Max.   :3.000   Max.   :6.000   Max.   :2590
##     Brick              Bedrooms       Bathrooms         Price
##  Length:128         Min.   :2.000   Min.   :2.000   Min.   : 69100
##  Class :character   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:111325
##  Mode  :character   Median :3.000   Median :2.000   Median :125950
##                     Mean   :3.023   Mean   :2.445   Mean   :130427
##                     3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:148250
##                     Max.   :5.000   Max.   :4.000   Max.   :211200
```

```
##
## Call:
## lm(formula = Price ~ . + Nbhd:Brick, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.01326 -0.19424 -0.01019  0.15994  1.02376
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.39326    0.07126  -5.519 2.05e-07 ***
## Nbhd2          -0.04904    0.09974  -0.492  0.62385
## Nbhd3           0.63199    0.12794   4.940 2.60e-06 ***
## Offers         -0.33358    0.04251  -7.846 2.15e-12 ***
## SqFt            0.42321    0.04477   9.453 3.96e-16 ***
## BrickYes        0.45008    0.15193   2.962  0.00369 **
## Bedrooms        0.12907    0.04286   3.011  0.00318 **
## Bathrooms       0.12365    0.04138   2.988  0.00341 **
## Nbhd2:BrickYes  0.09931    0.18865   0.526  0.59957
## Nbhd3:BrickYes  0.44413    0.19878   2.234  0.02735 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3665 on 118 degrees of freedom
## Multiple R-squared:  0.8752, Adjusted R-squared:  0.8657
## F-statistic: 91.94 on 9 and 118 DF,  p-value: < 2.2e-16
```

```
##                      2.5 %     97.5 %
## (Intercept)    -0.53437156 -0.2521460
## Nbhd2          -0.24654969  0.1484688
## Nbhd3           0.37863859  0.8853416
## Offers         -0.41776769 -0.2493884
## SqFt            0.33454792  0.5118642
## BrickYes        0.14921609  0.7509409
## Bedrooms        0.04419449  0.2139510
## Bathrooms       0.04170856  0.2055841
## Nbhd2:BrickYes -0.27427154  0.4728999
## Nbhd3:BrickYes  0.05048646  0.8377713
```

```
## Warning in abline(lm.fit): only using the first two of 10 regression
```

```
## coefficients
```




```
## [1] 0.1238296
```

MSE is 0.1238296

Part A

There is a premium for Brick houses since the co-efficient of the slope for brick houses is 0.45 and intercept is -0.39. The 95% confidence interval is [0.1492,0.7509], which does not include 0.

Part B

There is a premium for houses in neighborhood 3 since the co-efficient of the slope for houses in Nbhd3 is 0.6319 and intercept is -0.39. The 95% confidence interval is [0.3786,0.8853] which does not include 0.

Part C

There is a premium for Brick houses in neighborhood 3 since the co-efficient of the slope for brick houses in Nbhd3 is 0.441 and intercept is -0.39. The 95% confidence interval is [0.05,0.837] which does not include 0.

Part D

The 95% confidence interval for Nbhd2 co-efficient being 0 is [0.05,0.837] which does include 0. Therefore, we can safely say that there is no impact for the Nbhd2 on the regression output, thus allowing Nbhd2 to be combined with Nbhd1.

Word Problem 3 - Crime Data

Part A

Every city has different circumstances which might lead to an increase in crime, and these circumstances act as variables which might have a positive or negative correlation with the crime rate which are not attributable to the policing measures in the city. Therefore, one cannot directly define the crime rate with police presence, since there is no definitive way to establish that the police rate is directly responsible for change in crime.

Part B

In the UPenn research, it is mentioned that the increase in policing rate was not a direct result of an increase in crime, but it was due to a random, unrelated factor i.e the terrorism threat level in Washington DC. With virtually no change in other variables (i.e number of tourists on an orange
alert day),there is a negative correlation between police rate and crime, which is a smaller decrease (-6.046) versus when not controlling for METRO ridership(-7.316).

Part C

The study decided to control for METRO ridership because the initial hypothesis was that METRO ridership would decrease on an orange alert day, which would consequently decrease the number of victims. Therefore, in order to better test the performance of the model, the METRO ridership would have to be fixed to measure the direct impact of policing presence on crime rate. As can be seen from the table, the crime rate has a negative coefficient, suggesting that there is a decrease in crime rate with an increase in policing rate.

Part D

From the first column of table 2, it is visible that the standard error for District 1 is 0.044 whereas that for other districts is higher at 0.455. Therefore, it is safe to conclude that policing rate has a higher impact on crime rate in district 1 because district 1 maybe be a high priority target for potential terror attacks, which leads to a higher level of vigilance as compared to the other districts.

Word Problem 4 - Neural Nets

```
## The following objects are masked from Boston (pos = 10):
##
##     age, black, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad,
##     rm, tax, zn

## The following objects are masked from Boston (pos = 19):
##
##     age, black, chas, crim, dis, indus, lstat, medv, nox, ptratio, rad,
##     rm, tax, zn
```

```
##       crim                zn             indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax           ptratio          black
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

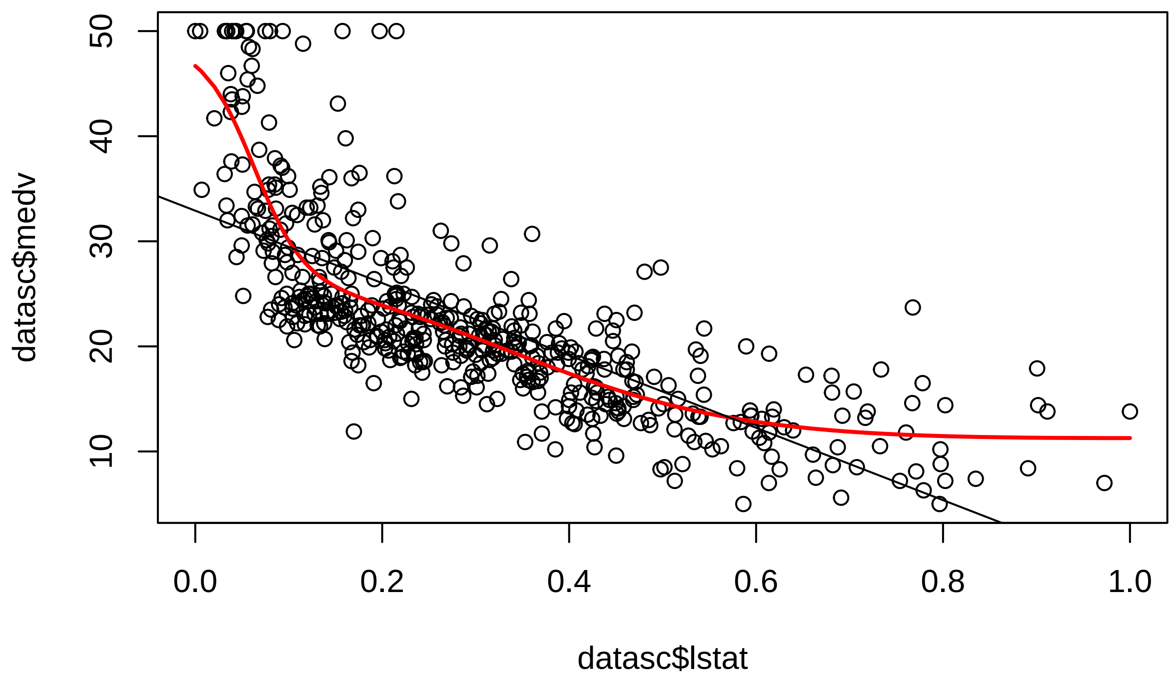
Part A

Using lstat as a variable to predict median housing value

```
## # weights:  10
## initial  value 306585.593206
## iter  10 value 26692.256860
## iter  20 value 15858.894040
## iter  30 value 14160.123240
## iter  40 value 13605.870335
## iter  50 value 13585.051877
## iter  60 value 13567.009598
## iter  70 value 13558.927242
## iter  80 value 13553.663537
## iter  90 value 13552.548083
## iter 100 value 13549.943379
## final  value 13549.943379
## stopped after 100 iterations
```

```
## a 1-3-1 network with 10 weights
## options were - linear output units  decay=0.1
##  b->h1 i1->h1
##   2.70  -7.97
##  b->h2 i1->h2
```

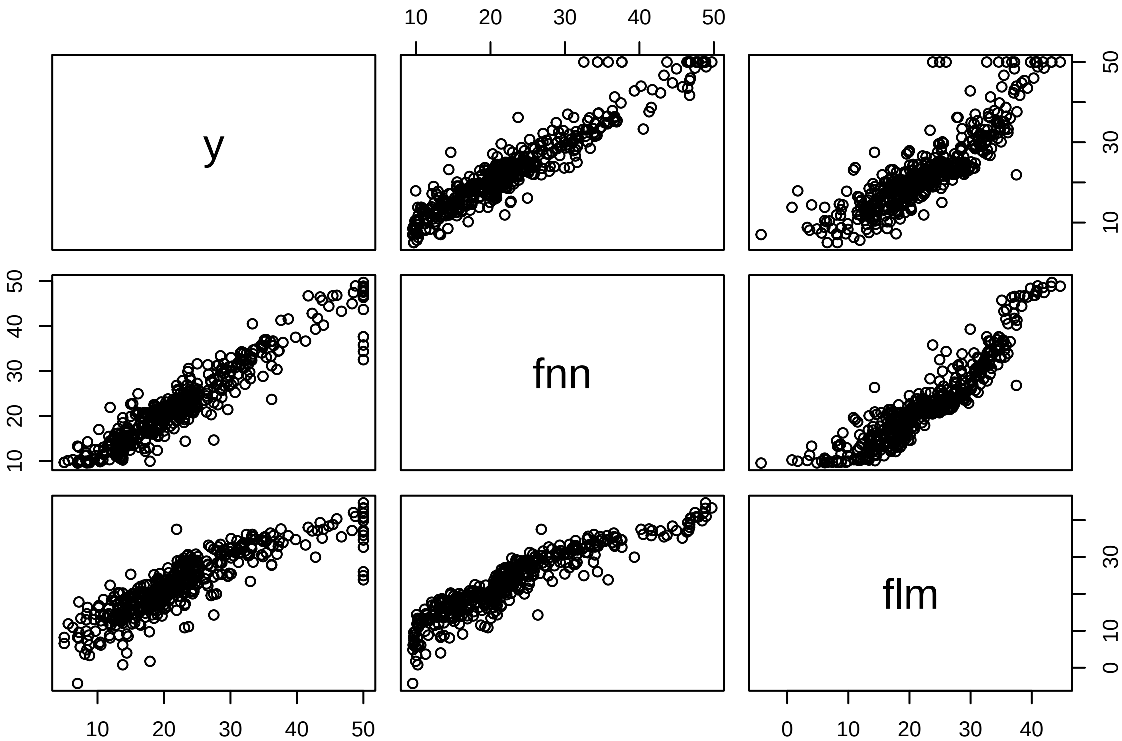
```
##   2.35 -37.50
##  b->h3 i1->h3
##  -1.08   1.67
##   b->o  h1->o  h2->o  h3->o
##   9.90  17.59  21.68   1.99
```



Part B

Running prediction for all variables

```
## # weights:  76
## initial  value 314133.312563
## iter  10 value 27319.520427
## iter  20 value 14627.345565
## iter  30 value 10978.215806
## iter  40 value 8851.783685
## iter  50 value 7939.947783
## iter  60 value 7491.251901
## iter  70 value 7099.260295
## iter  80 value 6320.436300
## iter  90 value 5945.687679
## iter 100 value 5551.555721
## final  value 5551.555721
## stopped after 100 iterations
```




```
##             y       fnn       flm
## y   1.0000000 0.9436769 0.8606060
## fnn 0.9436769 1.0000000 0.9148363
## flm 0.8606060 0.9148363 1.0000000
```

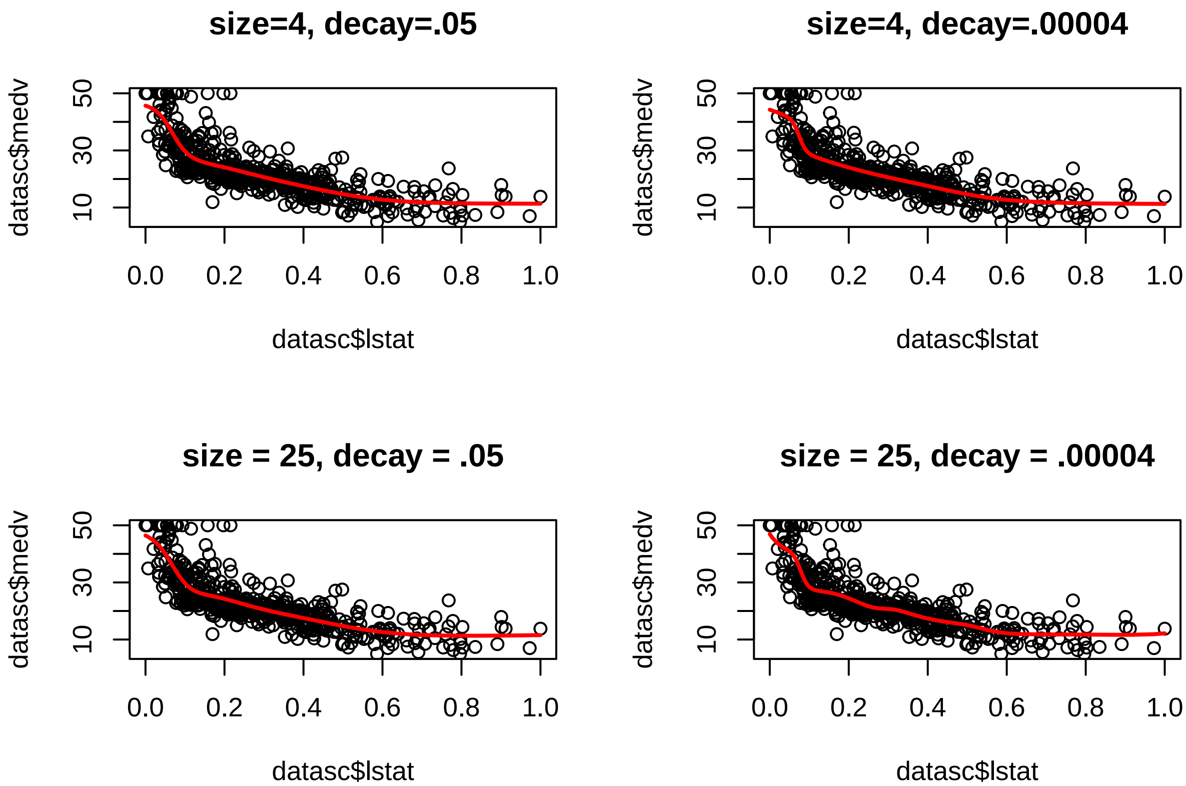
Part C

Size and Decay - four different fits

```
## # weights:  13
## initial  value 300433.769837
## iter  10 value 19970.625899
## iter  20 value 14594.224090
## iter  30 value 13692.480208
## iter  40 value 13587.563051
## iter  50 value 13506.348982
## iter  60 value 13416.409103
## iter  70 value 13412.518579
## iter  80 value 13411.434345
## iter  90 value 13409.522343
## iter 100 value 13408.083037
## final  value 13408.083037
## stopped after 100 iterations
```

```
## # weights:  13
```

```
## initial  value 275295.438735
## iter  10 value 18171.324798
## iter  20 value 13619.991719
## iter  30 value 13204.144567
## iter  40 value 13185.441003
## iter  50 value 13180.030952
## iter  60 value 13179.909094
## iter  70 value 13179.781949
## iter  80 value 13179.758591
## final  value 13179.755419
## converged

## # weights:  76
## initial  value 327768.695821
## iter  10 value 17864.597287
## iter  20 value 13967.553042
## iter  30 value 13777.126410
## iter  40 value 13664.618049
## iter  50 value 13577.456499
## iter  60 value 13520.859000
## iter  70 value 13497.779713
## iter  80 value 13485.642535
## iter  90 value 13475.047210
## iter 100 value 13468.592760
## final  value 13468.592760
## stopped after 100 iterations

## # weights:  76
## initial  value 330094.587354
## iter  10 value 15637.505467
## iter  20 value 13743.108614
## iter  30 value 13411.510459
## iter  40 value 13283.096662
## iter  50 value 13183.379730
## iter  60 value 13177.820936
## iter  70 value 13172.957924
## iter  80 value 13167.187225
## iter  90 value 13138.104286
## iter 100 value 13120.578575
## final  value 13120.578575
## stopped after 100 iterations
```



Plots

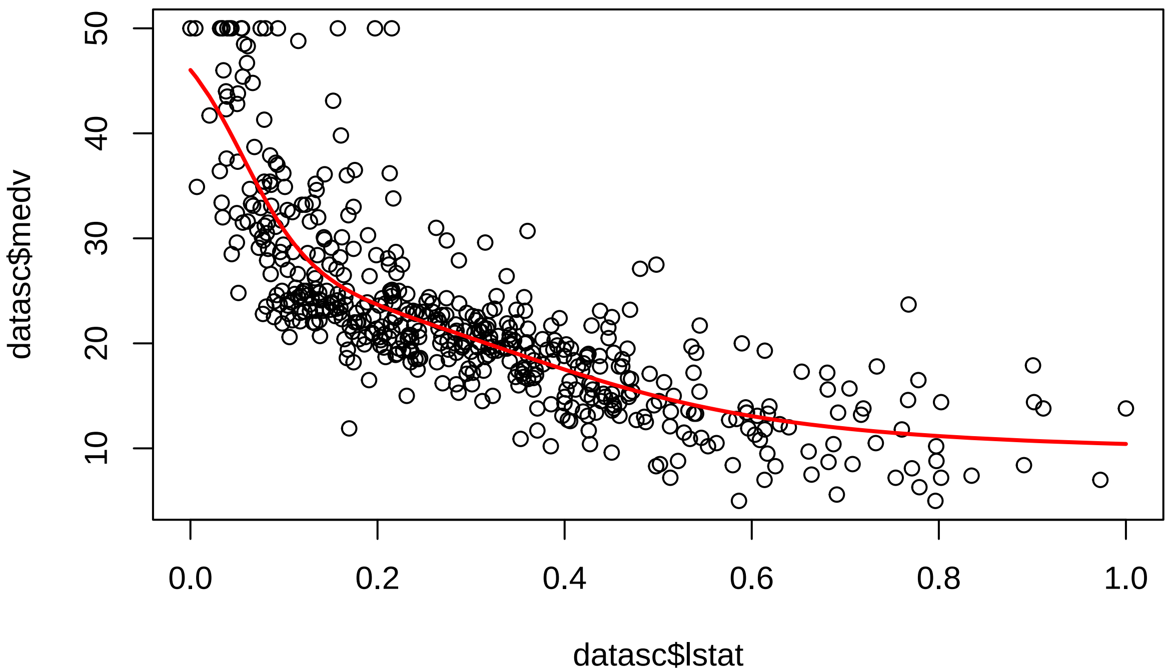
Part D

Iterative Fitting and Random Starting Values

```
## # weights:  151
## initial  value 289646.274673
## iter  10 value 15831.766825
## iter  20 value 14782.634427
## iter  30 value 14628.522271
## iter  40 value 14499.236834
## iter  50 value 14402.999950
## iter  60 value 14363.307762
## iter  70 value 14328.100140
## iter  80 value 14302.515814
## iter  90 value 14286.449515
## iter 100 value 14273.084058
## final  value 14273.084058
## stopped after 100 iterations
```

```
## # weights:  151
## initial  value 405441.811692
## iter  10 value 15527.299581
## iter  20 value 14611.580165
## final  value 14611.580165
## stopped after 20 iterations
```

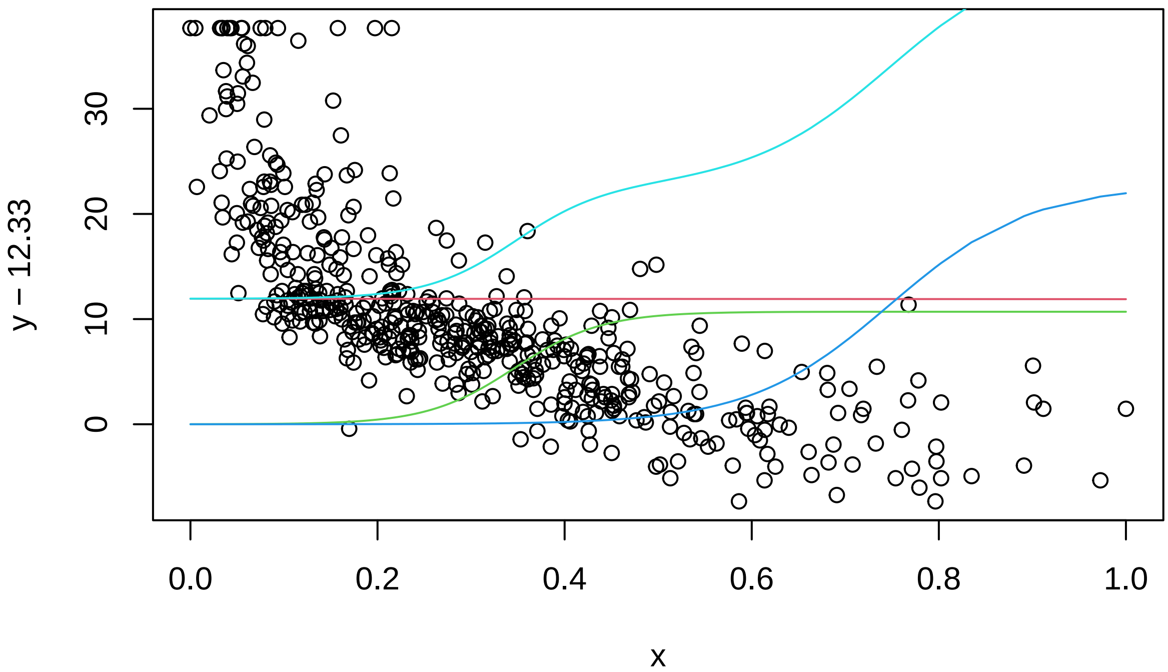
```
## # weights:  151
## initial  value 265868.763424
## iter  10 value 15852.644204
## iter  20 value 14986.076111
## iter  30 value 14794.995980
## iter  40 value 14601.168805
## iter  50 value 14471.792241
## iter  60 value 14393.111015
## iter  70 value 14325.237832
## iter  80 value 14300.591096
## iter  90 value 14286.214974
## iter 100 value 14277.820514
## iter 110 value 14270.699179
## iter 120 value 14266.354349
## iter 130 value 14264.046229
## iter 140 value 14261.657757
## iter 150 value 14258.339555
## iter 160 value 14255.983661
## iter 170 value 14253.213609
## iter 180 value 14250.305132
## iter 190 value 14248.503779
## iter 200 value 14247.297665
## iter 210 value 14246.034444
## iter 220 value 14245.198707
## iter 230 value 14244.657674
## iter 240 value 14244.078172
## iter 250 value 14243.762865
## iter 260 value 14243.525908
## iter 270 value 14243.252802
## iter 280 value 14243.000895
## iter 290 value 14242.734168
## iter 300 value 14242.568409
## iter 310 value 14240.584185
## iter 320 value 14239.449235
## final  value 14239.418499
## converged
```

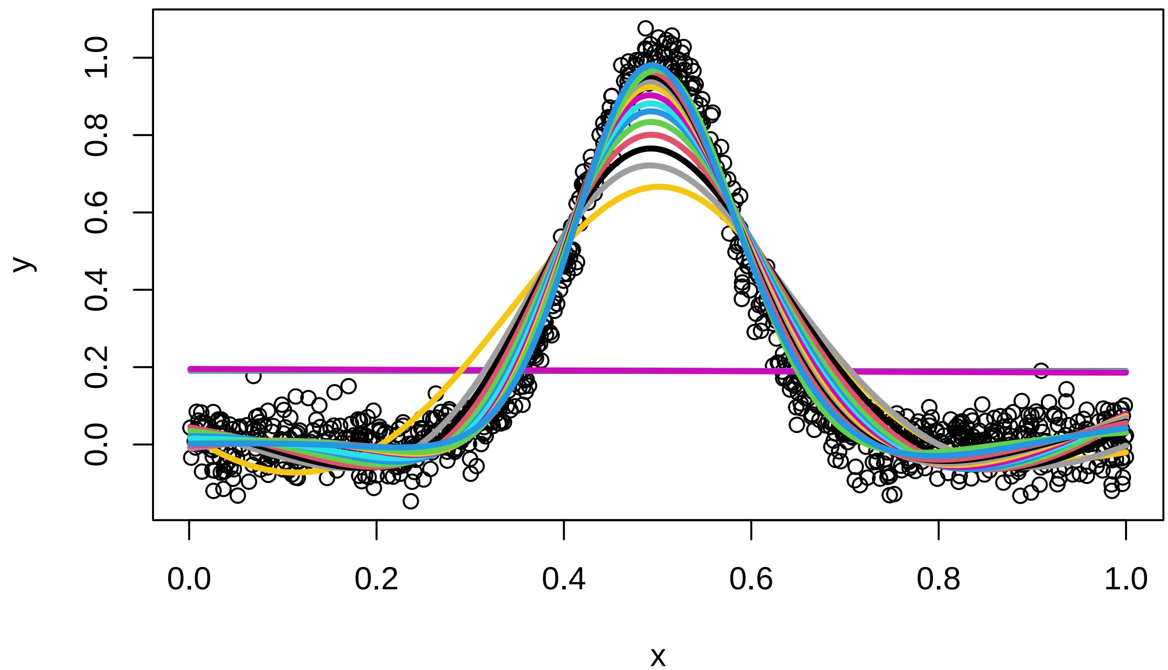


Part E

```
## a 13-5-1 network with 76 weights
## options were - linear output units  decay=0.1
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9->h1
##   -3.20    0.16   -4.58    8.13   -1.03   -3.21   36.71  -14.52  -14.70  -11.26
## i10->h1 i11->h1 i12->h1 i13->h1
##    5.32  -21.89   -2.83  -13.87
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2  i9->h2
##   -5.85    6.88   -8.18    1.86   10.46   11.36  -29.12   14.23  -10.15   12.78
## i10->h2 i11->h2 i12->h2 i13->h2
##   10.97    4.60  -31.88   27.23
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3  i9->h3
##   -1.58    0.23   -3.19    5.87   -1.40   -2.56   14.02    1.26    7.04   -6.62
## i10->h3 i11->h3 i12->h3 i13->h3
##  -10.30   -3.49   -6.15  -10.00
##   b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4  i9->h4
##   -1.52   -3.57    1.80   -2.13    0.72    1.72    5.97   -0.93   -5.48    4.19
## i10->h4 i11->h4 i12->h4 i13->h4
##   -0.69   -0.09    0.17  -10.67
##   b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5  i9->h5
##   13.02   -5.22   13.63    2.28    0.19   -8.21   -3.12   -0.62   -9.39    1.64
## i10->h5 i11->h5 i12->h5 i13->h5
##   -4.59   -4.62    1.80   -6.12
##   b->o  h1->o  h2->o  h3->o  h4->o  h5->o
```

```
##    6.74   6.04   2.79   8.43  15.43  13.88
```






```
## # weights:  10
## initial  value 667.209484
## iter  10 value 101.026184
## iter  20 value 100.820012
## final  value 100.810813
## converged
## [1] 1
##
## # weights:  10
## initial  value 102.440980
## iter  10 value 100.826954
## iter  20 value 100.807261
## final  value 100.806135
## converged
## [1] 2
##
## # weights:  10
## initial  value 541.209225
## iter  10 value 100.852025
## iter  20 value 100.816299
## iter  30 value 100.802004
## final  value 100.801949
## converged
## [1] 3
##
## # weights:  10
```

```
## initial  value 788.096197
## iter  10 value 101.194838
## iter  20 value 100.915328
## iter  30 value 100.806993
## iter  40 value 100.797891
## final  value 100.797834
## converged
## [1] 4
##
## # weights:  10
## initial  value 478.117642
## iter  10 value 101.685651
## iter  20 value 100.832005
## iter  30 value 100.794559
## iter  40 value 100.793583
## final  value 100.793575
## converged
## [1] 5
##
## # weights:  10
## initial  value 146.845885
## iter  10 value 100.818033
## iter  20 value 100.796759
## iter  30 value 100.790239
## iter  40 value 100.789554
## iter  50 value 100.789452
## iter  50 value 100.789451
## iter  50 value 100.789451
## final  value 100.789451
## converged
## [1] 6
##
## # weights:  10
## initial  value 329.771585
## iter  10 value 100.844858
## iter  20 value 97.533692
## iter  30 value 75.549916
## iter  40 value 65.922406
## iter  50 value 59.084955
## iter  60 value 56.115797
## iter  70 value 55.842127
## iter  80 value 55.639858
## iter  90 value 55.605056
## iter 100 value 55.599449
## iter 110 value 55.593087
## iter 120 value 55.580613
## iter 130 value 55.514138
## iter 140 value 54.092445
## iter 150 value 53.586253
## iter 160 value 53.226897
## final  value 53.226883
## converged
## [1] 7
##
```

```
## # weights:  10
## initial  value 3068.426044
## iter  10 value 100.371708
## iter  20 value 68.246061
## iter  30 value 48.625330
## iter  40 value 45.276644
## iter  50 value 44.401786
## iter  60 value 43.643691
## iter  70 value 43.471197
## iter  80 value 43.345269
## iter  90 value 43.337162
## iter  90 value 43.337162
## iter  90 value 43.337162
## final  value 43.337162
## converged
## [1] 8
##
## # weights:  10
## initial  value 475.529741
## iter  10 value 95.725015
## iter  20 value 60.553867
## iter  30 value 44.925104
## iter  40 value 41.677244
## iter  50 value 39.739201
## iter  60 value 38.905639
## iter  70 value 37.649659
## iter  80 value 36.193263
## iter  90 value 35.941581
## iter 100 value 35.937974
## final  value 35.937970
## converged
## [1] 9
##
## # weights:  10
## initial  value 1088.741534
## iter  10 value 97.327375
## iter  20 value 62.897333
## iter  30 value 58.408524
## iter  40 value 45.125803
## iter  50 value 34.710876
## iter  60 value 31.570089
## iter  70 value 30.069848
## iter  80 value 29.890360
## iter  90 value 29.787805
## iter 100 value 29.786191
## final  value 29.786124
## converged
## [1] 10
##
## # weights:  10
## initial  value 123.832237
## iter  10 value 93.503417
## iter  20 value 55.730199
## iter  30 value 45.149636
```

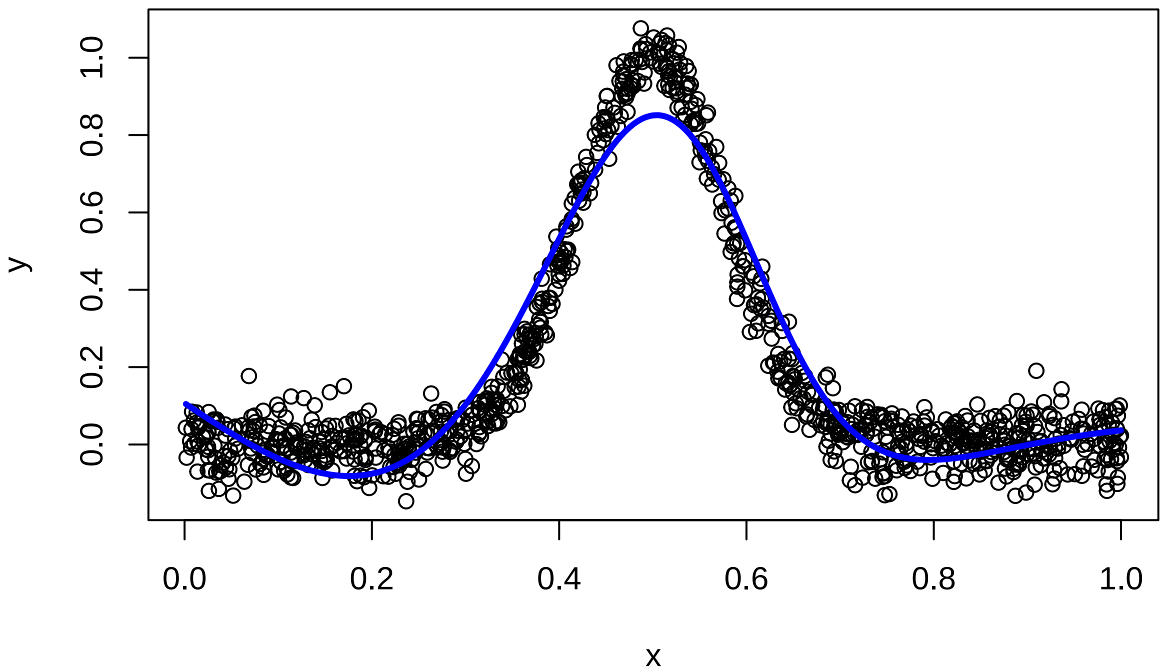
```
## iter  40 value 37.068339
## iter  50 value 26.394398
## iter  60 value 25.198380
## iter  70 value 24.821157
## iter  80 value 24.786035
## final  value 24.785776
## converged
## [1] 11
##
## # weights:  10
## initial  value 941.552226
## iter  10 value 100.917847
## iter  20 value 95.645178
## iter  30 value 40.872195
## iter  40 value 24.362123
## iter  50 value 21.576139
## iter  60 value 20.638054
## iter  70 value 20.375986
## iter  80 value 20.258902
## iter  90 value 20.237569
## iter 100 value 20.221741
## final  value 20.221565
## converged
## [1] 12
##
## # weights:  10
## initial  value 100.954276
## iter  10 value 100.784493
## iter  20 value 94.318711
## iter  30 value 59.559869
## iter  40 value 43.853709
## iter  50 value 30.149774
## iter  60 value 21.512915
## iter  70 value 19.926423
## iter  80 value 18.164267
## iter  90 value 17.836702
## iter 100 value 17.810040
## final  value 17.808814
## converged
## [1] 13
##
## # weights:  10
## initial  value 182.125022
## iter  10 value 95.939831
## iter  20 value 29.937943
## iter  30 value 19.496491
## iter  40 value 17.283913
## iter  50 value 16.454872
## iter  60 value 15.330165
## iter  70 value 14.720350
## iter  80 value 14.237581
## iter  90 value 14.222351
## iter 100 value 14.210798
## iter 110 value 14.210085
```

```
## final  value 14.210020
## converged
## [1] 14
##
## # weights:  10
## initial  value 207.261262
## iter  10 value 100.794214
## iter  20 value 97.896878
## iter  30 value 67.777669
## iter  40 value 36.111341
## iter  50 value 25.592396
## iter  60 value 12.103487
## iter  70 value 11.750128
## iter  80 value 11.617151
## iter  90 value 11.598152
## iter 100 value 11.584226
## final  value 11.583959
## converged
## [1] 15
##
## # weights:  10
## initial  value 104.282101
## iter  10 value 96.826786
## iter  20 value 54.705298
## iter  30 value 40.571252
## iter  40 value 36.439169
## iter  50 value 33.754560
## iter  60 value 17.557532
## iter  70 value 10.553777
## iter  80 value 9.913764
## iter  90 value 9.724173
## iter 100 value 9.689113
## iter 110 value 9.686717
## final  value 9.686504
## converged
## [1] 16
##
## # weights:  10
## initial  value 107.517037
## iter  10 value 99.179407
## iter  20 value 74.103260
## iter  30 value 39.576816
## iter  40 value 35.584109
## iter  50 value 34.860369
## iter  60 value 34.531369
## iter  70 value 33.457824
## iter  80 value 30.277923
## iter  90 value 14.212284
## iter 100 value 9.321129
## iter 110 value 9.152853
## iter 120 value 8.948148
## iter 130 value 8.925002
## iter 140 value 8.812747
## iter 150 value 8.775124
```

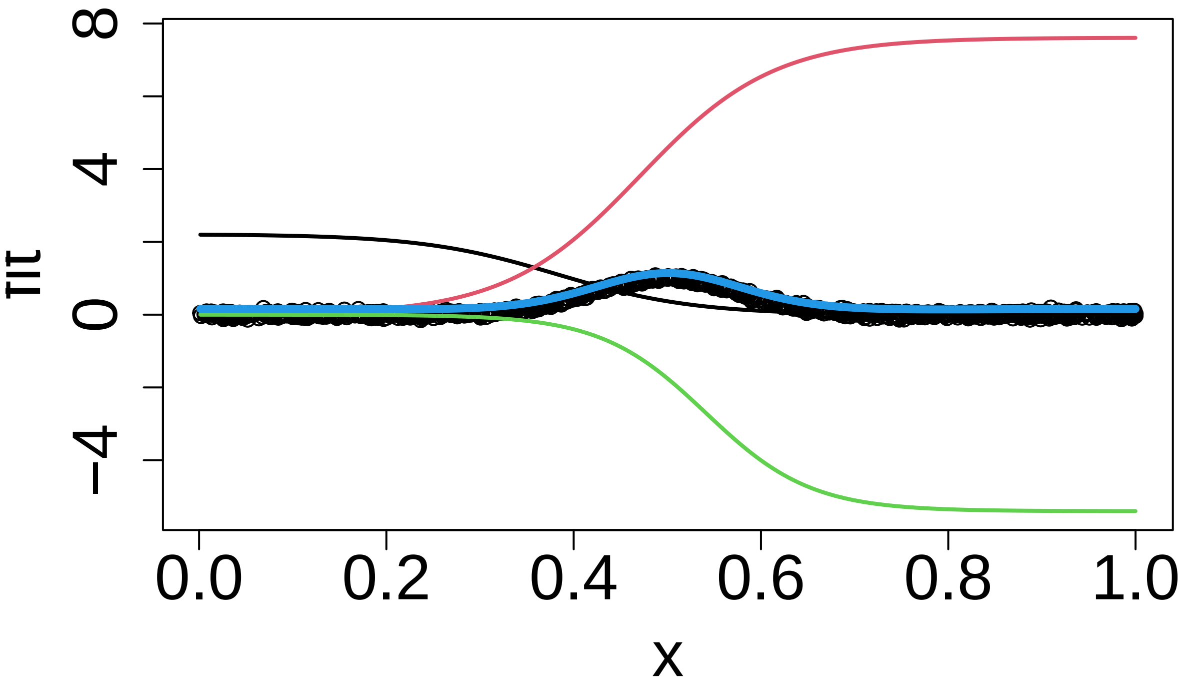
```
## iter 160 value 8.693114
## iter 170 value 8.676177
## iter 180 value 8.611530
## iter 190 value 8.573500
## iter 200 value 8.529510
## iter 210 value 8.524129
## iter 220 value 8.523541
## final  value 8.523484
## converged
## [1] 17
##
## # weights:  10
## initial  value 532.748497
## iter  10 value 100.461278
## iter  20 value 29.100384
## iter  30 value 8.345113
## iter  40 value 8.237850
## iter  50 value 7.803442
## iter  60 value 7.515991
## iter  70 value 7.364291
## iter  80 value 7.349896
## iter  90 value 7.328506
## iter 100 value 7.315689
## iter 110 value 7.296771
## iter 120 value 7.291703
## iter 130 value 7.276114
## iter 140 value 7.258357
## iter 150 value 7.242286
## iter 160 value 7.217620
## iter 170 value 7.185886
## iter 180 value 7.146095
## iter 190 value 7.140940
## iter 200 value 7.140355
## final  value 7.140350
## converged
## [1] 18
##
## # weights:  10
## initial  value 209.820417
## iter  10 value 100.728480
## iter  20 value 93.561223
## iter  30 value 66.784151
## iter  40 value 14.614305
## iter  50 value 10.118364
## iter  60 value 7.581487
## iter  70 value 7.116772
## iter  80 value 6.847553
## iter  90 value 6.559304
## iter 100 value 6.388675
## iter 110 value 6.275630
## iter 120 value 6.220974
## iter 130 value 6.174280
## iter 140 value 6.152152
## iter 150 value 6.132159
```

```
## iter 160 value 6.123116
## iter 170 value 6.062432
## iter 180 value 6.042175
## iter 190 value 6.033226
## iter 200 value 6.032984
## final  value 6.032940
## converged
## [1] 19
##
## # weights:  10
## initial  value 173.792531
## iter  10 value 100.754535
## iter  20 value 89.667530
## iter  30 value 39.589508
## iter  40 value 7.077317
## iter  50 value 5.799415
## iter  60 value 5.684139
## iter  70 value 5.558442
## iter  80 value 5.477864
## iter  90 value 5.440157
## iter 100 value 5.423128
## iter 110 value 5.419010
## iter 120 value 5.417217
## final  value 5.417011
## converged
## [1] 20
```

```
## # weights:  10
## initial  value 465.933841
## iter  10 value 100.787554
## iter  20 value 98.638855
## iter  30 value 64.470715
## iter  40 value 53.295894
## iter  50 value 39.429427
## iter  60 value 25.916256
## iter  70 value 25.174776
## iter  80 value 22.051468
## iter  90 value 21.301871
## iter 100 value 21.102706
## iter 110 value 21.085204
## iter 120 value 21.075905
## final  value 21.075860
## converged
```



y
x
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0



Word Problem 5 - Project Contribution

Contributions to project

In my group project, I began by taking the initiative to search for relevant datasets to explore and model. After searching through Kaggle, my teammates and I settled on an HR analytics project to study the efficacy and budget for data science related trainings. Simultaneously, I started having discussions with each of my teammates separately in order to better understand their grasp on R and ML concepts along with their presentation skills. This proved to be a very impactful step, which allowed us to better streamline our approach.

We then held a brief meeting, where I used my understanding of my teammates' abilities to put forth a division of labor plan, which best played to each of my teammate's strengths. I then began work on a powerpoint presentation early so as to ensure that we weren't underprepared a day before the presentation. I also went through the Chapter 8 lab in the book step-by-step with my team, to better understand decision trees and began work on modelling the dataset using Random Forests. Additionally, I assisted with Exploratory Data Analysis on the dataset, cleaning the data from relevant blanks and interpreting data insights via pivot tables and Pareto charts to include into the presentation. Once the EDA and modelling was complete, I assisted in cross checking my teammates' codes within the group to ensure consistency and accuracy throughout our models.

Once time for presentation preparation arrived, my teammate's and I were able to select our presentation slides and talking points with relative ease due to labour division. However, we made it a point to know each other's source material thoroughly to ensure that we deliver a smooth and cohesive presentation. I also added relevant visuals to the powerpoint in conjunction with my team, and we were satisfied with the impact the presentation had on the audience due to the combination of above factors. I learned a lot about R modelling and presentation delivery along with teamwork skills in this presentation, which I hope to cultivate throughout my MSBA and carry over into my professional endeavors.